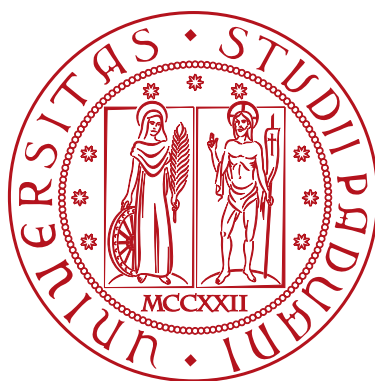




UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea

La metodologia SCRUM nello sviluppo di prodotti tangibili: il caso FILA

Relatore

Ch.mo Prof. Stefano Biazzo

Correlatore

Ing. Alessandro Ruta

Laureando

Compagno Daniel

Anno Accademico 2019-2020

*“Tieni sempre presente che la tua ferma
convinzione di riuscire è più importante
di qualsiasi altra cosa.”*

Abramo Lincoln

Dedico il raggiungimento di questo importante traguardo alla mia famiglia, che con sacrifici lo ha reso realizzabile e alla quale non sarò mai abbastanza grato. Ai miei amici, per avermi supportato e sopportato, soprattutto quando lo studio e il lavoro hanno avuto la priorità.

Ad Elena, per essere la mia àncora, sia nei momenti di gioia che in quelli più difficili.

A tutto il team di auxiell per essere stato un esempio ed un motivo di crescita in questi mesi, in particolare a Giacomo e Alessandro per avermi accompagnato da vicino in questo percorso. Al Ch.mo prof. Biazzo per la gentilezza e la disponibilità dimostrata. A chi non c'è più, ma mi guarda dall'alto. A tutte le persone che ho incontrato in questo lungo viaggio e che mi hanno reso la persona che sono oggi.

Sommario

Il processo di *Development*, assieme ai processi di *Delivery*, *Deployment* e *Demand*, rappresenta uno dei 4 macro-processi di generazione del valore all'interno di un'organizzazione.

In (Morgan & Liker, 2006), viene evidenziato come “*Ci sono più opportunità di vantaggio competitivo nel processo di sviluppo che in qualsiasi altro luogo*”, sottolineando l'importanza di efficientare tale processo per la crescita di un'azienda.

La gestione del processo di sviluppo rappresenta però una sfida all'interno delle organizzazioni, le quali devono essere in grado di fronteggiare un mercato in rapida evoluzione, clienti sempre più esigenti e *time to market* sempre più stringenti.

Tale problematica viene affrontata nel mondo dello sviluppo software a partire dagli anni '90, portando alla luce diverse metodologie: alla conferenza OOPSLA¹ del 1995 Jeff Sutherland e Ken Shwaber formalizzano *Scrum*, un “*modo più veloce, affidabile ed efficace di creare software per il settore tecnologico*” (Sutherland, 2015).

Da quel momento ad oggi, *Scrum* è diventato uno dei più famosi *framework* per la gestione dei progetti di sviluppo software. Nonostante Jeff Sutherland nel suo libro “*Fare il doppio in metà tempo*” ne sottolinei i benefici nei più svariati campi applicativi, l'implementazione del framework non ha avuto la stessa risonanza all'interno dell'industria dello sviluppo hardware.

Gli obiettivi che questa tesi si pone sono analizzare se il framework *Scrum* possa essere implementato con successo nel processo di sviluppo di prodotti tangibili e valutare la necessità di modificarne alcune caratteristiche per adattarlo al contesto.

Nella prima parte dell'elaborato è esposta un'analisi di *Scrum* dal punto di vista teorico e le evidenze presenti in letteratura riguardo l'implementazione nello sviluppo di prodotti tangibili, con la considerazione di approcci ibridi e derivati.

Nella seconda parte è invece descritta e analizzata l'implementazione in FILA Industria Chimica s.p.a. e FreTor s.r.l., aziende che stanno seguendo un percorso di *Complete Lean Transformation* supportate da *auxiell*, società di consulenza *lean* con la quale ho svolto il mio tirocinio formativo.

¹ *Object-Oriented Programming, Systems, Languages & Applications.*

Indice

Introduzione	1
Capitolo 1. Scrum.....	5
<i>1.1 Le origini</i>	<i>6</i>
<i>1.2 Il Manifesto Agile per lo Sviluppo Software</i>	<i>10</i>
<i>1.3 Il Framework Scrum</i>	<i>12</i>
1.3.1 La teoria di Scrum	13
1.3.2 I Ruoli di Scrum	14
1.3.3 Lo Sprint	18
1.3.4 Gli Artefatti di Scrum	19
1.3.5 Gli Eventi di Scrum	24
Capitolo 2. Scrum nello sviluppo Hardware.....	29
<i>2.1 Le sfide dello sviluppo hardware con Scrum.....</i>	<i>30</i>
<i>2.2 Extreme Manufacturing.....</i>	<i>33</i>
<i>2.3 Agile Stage-Gate</i>	<i>36</i>
2.3.1 La gestione dei progetti tramite Hybrid Stage-Gate	38
2.3.2 Benefici e sfide dei modelli ibridi.....	39
2.3.3 Caratteristiche dei processi di sviluppo con modelli ibridi	42
2.3.4 Criteri flessibili ai Gate.....	43
Capitolo 3. Il Caso studio: FILA Industria Chimica s.p.a.	45
<i>3.1 auxiell s.p.a</i>	<i>46</i>
<i>3.2 Introduzione a FILA Industria Chimica s.p.a.</i>	<i>48</i>
3.2.1 L'offerta di FILA.....	49
3.2.2 La ricerca e sviluppo.....	49
3.2.3 La necessità di evolvere	50
3.2.3 Lean Gap Analysis® e Lean RoadMap®	51
3.2.3.1 Value Stream Mapping.....	51
3.2.3.2 Definizione del progetto pilota.....	52
3.2.3.3 Current State Map	53
3.2.3.4 Muda Analysis.....	54
3.2.3.5 Future State Map	57
<i>3.3 Organizzazione del processo.....</i>	<i>61</i>
<i>3.4 Formalizzazione degli standard di Scrum in FILA.....</i>	<i>64</i>
3.4.1 I Ruoli	64
3.4.2 Gli Eventi	66

3.5	<i>Esempio di svolgimento di uno Sprint in FILA</i>	77
Capitolo 4.	Il Caso studio: FreTor s.r.l.	83
4.1.	<i>Introduzione a FreTor s.r.l.</i>	84
4.2.	<i>Organizzazione del processo.....</i>	85
4.2.1	<i>Attività preparatorie alla pianificazione con Scrum.....</i>	88
4.3	<i>Esempio di svolgimento di uno Sprint in FreTor.....</i>	92
Capitolo 5.	Modello di implementazione, Analisi critica e Conclusioni.....	99
5.1	<i>Analisi dei casi studio.....</i>	100
5.2	<i>Modello di implementazione.....</i>	105
5.3	<i>Difficoltà nell'utilizzo di Scrum.....</i>	107
5.4	<i>Adattamenti nell'utilizzo di Scrum.....</i>	109
5.5	<i>Conclusioni.....</i>	111
Bibliografia		113
	<i>Indice delle Figure</i>	115

Introduzione

Il *Project Management Institute*², all'interno della *PMBOK® Guide*, definisce il ciclo di vita di un progetto come:

“the series of phases that a project passes through from its initiation to its closure”.

(Project Management Institute, 2013)

L'approccio tradizionale alla gestione del ciclo di vita dei progetti prevede l'applicazione di una metodologia *plan-driven*, ovvero:

“Predictive life cycles (also known as fully plan-driven) are ones in which the project scope, and the time and cost required to deliver that scope, are determined as early in the project life cycle as practically possible.”

(Project Management Institute, 2013)

In accordo con (Morgan & Keshta, 2017), le caratteristiche della metodologia *plan-driven* sono le seguenti:

- **Approccio predittivo:** la maggior parte del progetto è pianificata e definita nei requisiti di progetto. La metodologia segue una sistematica disciplina che la rende applicabile a progetti prevedibili e ripetibili.
- **Documentazione completa:** l'approccio tradizionale pone una particolare attenzione alla fase di definizione dei requisiti, nella fase iniziale del progetto; tutti i requisiti, input, output e obiettivi del cliente, devono essere documentati prima di iniziare qualsiasi attività. Spesso coloro che partecipano alla definizione dei requisiti non sono gli stessi che eseguono le attività, quindi è necessario rendere specifico cosa ci si aspetta venga realizzato in un documento.
- **Process-oriented:** il manager e le altre figure impegnate devono attenersi continuamente alle attività/fasi del progetto che sono state definite nella fase di pianificazione; l'orientamento al processo risulta la caratteristica predominante dell'approccio tradizionale.

² Il Project Management Institute è un'organizzazione professionale globale no-profit per la gestione dei progetti.

I principi caratterizzanti la metodologia plan-driven hanno trovato applicazione in diversi modelli all'interno dell'ambito del processo di sviluppo prodotto.

Il modello *Stage-Gate*, formalizzato da Robert G. Cooper³ negli anni '80, risulta uno degli approcci più adottati nello sviluppo di prodotti tangibili. Tale modello prevede la suddivisione del processo di sviluppo in Gate, ovvero momenti decisionali, e Stage, l'insieme delle attività predefinite che precedono il gate.

Nel mondo dello sviluppo software si è diffuso il modello *waterfall*, termine utilizzato per la prima volta da Bell e Thayer in un articolo del 1976, ad indicare un processo in cui ciascuna fase produce un ben preciso output (deliverable) che viene utilizzato come input per la fase successiva.

Alcuni dei principali limiti dei modelli fondati sulla metodologia plan-driven, in accordo con (Morgan & Keshta, 2017), risultano essere:

- Rigidità alle esigenze mutevoli;
- Supporto limitato per progetti esplorativi in ambienti che cambiano;
- Team di sviluppo che non possiedono la libertà di prendere decisioni, ovvero devono seguire la documentazione;
- Coinvolgimento dei clienti limitato alla definizione dei requisiti;
- Coinvolgimento dei clienti che avviene solamente attraverso mezzi formali.

(Cooper R. G., 2014) sostiene che: “il contesto nel quale nascono e trovano applicazione tali modelli è estremamente diverso rispetto all'ambiente attuale, il quale risulta più veloce, più competitivo e globale e meno prevedibile.”

³ Dr. Robert G. Cooper è il creatore del modello Stage-Gate. È stato nominato “World’s Top Innovation Management Scholar” dal prestigioso U.S. Journal of Product Innovation Management ed è un socio del Product Development and Management Association (PDMA).

Con un'accezione più generica, possiamo descrivere il contesto attuale con l'acronimo VUCA, termine utilizzato per la prima volta nel 1986 da Warren Bennis⁴ e Burt Nanus⁵, e che ha preso piede all'interno dell'ambiente organizzativo a partire dagli anni 2000.

VUCA fa riferimento a 4 condizioni (Complexity Education Project, 2018):

Volatility. Fluttuazioni, turbolenze e rapide evoluzioni hanno un impatto determinante nella natura dei cambiamenti in un determinato contesto, in particolar modo negli ambienti economici. Tanto più un determinato contesto è volatile e tanto più i cambiamenti sono rapidi.

Uncertainty. La condizione di conoscenza non perfetta comporta una mancanza di sicurezza riguardo a cosa sta succedendo nell'ambiente in cui siamo inseriti, nell'interpretazione dei fenomeni in atto e nella formulazione di previsioni attendibili sul futuro. Più il mondo è incerto, più è difficile da prevedere.

Complexity. Diversi fattori possono interagire e influenzare le dinamiche all'interno di un determinato contesto; inoltre, tanto più questi fattori sono numerosi, diversi tra loro e interconnessi e tanto più il contesto che influenzano risulta complesso. Come conseguenza, anche l'analisi e lo studio di tale contesto diventano estremamente complessi.

Ambiguity. Informazioni incomplete, contraddittorie o non completamente corrette comportano una difficoltà nell'interpretare adeguatamente un determinato fenomeno.

Nello specifico dell'ambito dello sviluppo prodotto, coerentemente con il termine VUCA, Preston G. Smith⁶ individua due ragioni per le quali è necessario modificare le metodologie con cui si gestisce tale processo. (Smith, 2007)

La prima ragione è che il mondo in cui viviamo è diventato molto più caotico, infatti, da un lato i clienti sono più sofisticati, più esigenti e quindi più facilitati a cambiare idea, e dall'altro il mercato è diventato più turbolento, con competitor che possono provenire da ogni angolo del globo. In secondo luogo, le modalità con cui vengono gestiti i progetti provocano rigidità, in quanto l'approccio di management tradizionale è orientato alla pianificazione di un progetto in anticipo e i team sono incoraggiati e ricompensati sulla "bravura" nel seguire tale piano.

⁴ Warren Gamaliel Bennis (8 marzo 1925 - 31 luglio 2014) era uno studioso americano, consulente organizzativo e autore, ampiamente considerato come un pioniere nel campo contemporaneo degli studi di Leadership.

⁵ Burton B. "Burt" Nanus (nato nel 1936, New York NY) è un accademico americano, professore emerito all'Università della California del sud, dove è stato membro di facoltà dal 1969 al 1994.

⁶ Autore di diversi libri incentrati sul processo di sviluppo prodotto, tra cui *Flexible Product Development: Building Agility for Changing*, 2017

L'analisi delle caratteristiche dei modelli tradizionali e del contesto attuale di riferimento ha permesso di comprendere l'inadeguatezza di tali metodologie a supporto della gestione del processo di sviluppo prodotto.

Nello sviluppo software, per far fronte alle scadenti performance dei progetti, a partire dagli anni '90 sono nate diverse metodologie, tra cui Scrum.

Dalla sua formalizzazione ad oggi Scrum è uno dei framework più utilizzati, sia dalle multinazionali (come Google, Yahoo, Siemens) che dalle piccole-medie imprese, per raggiungere elevate performance nello sviluppo di prodotti software. È stato adottato per realizzare il prodotto software più redditizio della storia, Google Adwords, e il più performante progetto di grosse dimensioni, SiriDynix, con oltre un milione di linee di codice (Sutherland, 2012).

Nonostante i principi cardine di Scrum non abbiano un legame diretto con l'industria dello sviluppo software e, di conseguenza, si possano considerare applicabili in qualsiasi contesto, è di pensiero comune che l'applicazione del framework sia confinata al contesto da cui ha avuto origine, infatti i riferimenti in letteratura e i casi studio analizzabili riguardo all'utilizzo di Scrum nello sviluppo di prodotti tangibili risultano scarsi.

Gli obiettivi che questa tesi si pone sono individuare se il framework Scrum possa essere utilizzato con successo anche nello sviluppo di prodotti tangibili e se è necessario adattare alcuni aspetti per fare in modo che si integri in questo contesto.

Il primo capitolo è dedicato all'analisi di letteratura di Scrum, dalle origini agli elementi fondamentali per l'implementazione.

Nel secondo capitolo è esposta un'analisi di letteratura riguardo l'applicazione del framework nello sviluppo di prodotti tangibili, i diversi approcci che sono nati per favorirne l'implementazione e le sfide all'interno di tale contesto.

Il terzo capitolo e il quarto capitolo sono dedicati a due casi studio aziendali, dove Scrum è stato adottato come framework di pianificazione per lo sviluppo di prodotti professionali per la pulizia e l'igiene e per lo sviluppo di macchine industriali automatizzate.

Infine, nell'ultimo capitolo dell'elaborato è presentata una proposta di modello di implementazione che comprende un diagramma di flusso in cui sono definite tutte le attività e gli standard per l'utilizzo del framework e successivamente sono esposte un'analisi critica dell'implementazione di Scrum in relazione alle evidenze presenti in letteratura e alcune considerazioni conclusive riguardo al lavoro svolto.

Capitolo 1.

Scrum

*“Il tempo è relativo, il suo unico valore è dato da ciò che noi
facciamo mentre sta passando.”*

Albert Einstein

L’obiettivo di questo capitolo è fornire una panoramica approfondita del framework Scrum da un punto di vista teorico, partendo dalle origini fino ad analizzare le sue componenti e le regole presenti in letteratura per la corretta implementazione.

1.1 Le origini

Nel 1986, Hirotaka Takeuchi⁷ e Ikujiro Nonaka⁸ pubblicano nell'edizione di gennaio dell'*Harvard Business Review* un articolo intitolato “*The New New Product Development Game*”⁹.

Nell'articolo i due ricercatori evidenziano come le aziende si stiano progressivamente rendendo conto che l'approccio tradizionale di sviluppo prodotto, definito sequenziale, non sia adatto al raggiungimento degli obiettivi richiesti dal mercato.

Al contrario, alcune aziende Giapponesi e Statunitensi stanno adottando un “*metodo olistico, in cui, come nel rugby, la palla viene passata all'interno della squadra mentre si muove come un'unità sul campo.*” (Takeuchi & Nonaka, 1986). Questo approccio risulta più adatto ai requisiti competitivi del mercato che, oltre a prevedere bassi costi, elevata qualità e differenziazione, si sta orientando verso una ricerca di velocità e flessibilità.

Takeuchi e Nonaka affermano che per muovere la mischia verso fondo campo (traduzione di “*Moving the Scrum*”¹⁰ *downfield*”) sia necessario un processo di sviluppo che presenti 6 caratteristiche (Takeuchi & Nonaka, 1986):

Instabilità incorporata. Il management non deve fornire un piano di lavoro specifico, ma indicare una direzione strategica generale o un obiettivo ampio, in modo da garantire al team di progetto ampia libertà e, allo stesso tempo, obiettivi sfidanti.

Team di progetto auto-organizzati. Il team viene fatto operare all'interno di un contesto in cui non viene sfruttata la conoscenza pregressa e

⁷ Nato nel 1946, Hirotaka Takeuchi si è laureato presso la *International Christian University* di Tokyo. Dopo un breve periodo di lavoro presso *McCann-Erickson*, si è trasferito all'Università della California, a Berkeley, dove ha conseguito l'MBA nel 1971 e il dottorato di ricerca nel 1977. Durante il suo periodo a Berkeley, ha lavorato presso la *McKinsey & Company* a Tokyo dove ha incontrato Nonaka. Takeuchi prese una cattedra ad *Harvard* dal 1976 fino al 1983, quando entrò a far parte della *School of Commerce* dell'Università di Hitotsubashi, dove divenne professore ordinario e Decano della *Graduate School of International Corporate Strategy*. Rimase fino al 2010, quando tornò ad Harvard, come professore di *Management Practice*, dove si trova ora.

⁸ Nato nel 1935, Ikujiro Nonaka ha conseguito una laurea in scienze politiche all'Università di Waseda, quindi ha iniziato a lavorare alla *Fuji Electric*, dove ha creato il loro programma di management. Lasciò *Fuji* nel 1967, per studiare all'Università della California, Berkeley. Ha conseguito l'MBA nel 1968 e il dottorato in Economia aziendale nel 1972. Ha assunto incarichi presso università statunitensi, prima di tornare in Giappone, come professore presso la *Graduate School of International Corporate Strategy* della *Hitotsubashi University*.

⁹ <https://hbr.org/1986/01/the-new-new-product-development-game>

¹⁰ Il termine *Scrum* viene utilizzato nell'articolo come metafora del gioco del Rugby. La mischia chiusa o mischia ordinata (*Scrum*, in lingua Inglese) è una situazione di gioco che si crea per ordine dell'arbitro per riprendere il gioco quando esso è stato interrotto per qualche irregolarità.

questo lo obbliga ad auto-organizzarsi. Il team di progetto opera come una start-up, assumendosi rischi, proponendo iniziative e sviluppando un proprio piano. Un team in grado di auto-organizzarsi possiede 3 caratteristiche: autonomia (il team è libero di impostare il proprio lavoro), trascendenza (il team possiede un senso di finalità che va oltre l'ordinario) e inter-funzionalità (il team possiede tutte le competenze per portare a termine il lavoro).

Fasi di sviluppo sovrapposte. L'approccio tradizionale è caratterizzato da fasi del ciclo di vita del progetto che si susseguono in modo sequenziale, passando alla fase successiva solamente se vengono soddisfatti tutti i requisiti della fase precedente. Se si presentassero dei colli di bottiglia, il processo di sviluppo potrebbe rallentare o addirittura bloccarsi. L'approccio olistico prevede una sovrapposizione delle fasi di sviluppo, in modo da assorbire le interferenze lungo il processo che, così facendo, non si interrompe.

Apprendimento multiplo. Questo termine fa riferimento a due diverse tipologie di apprendimento che vengono incoraggiate nelle organizzazioni: apprendimento multilivello e apprendimento inter-funzionale. Con apprendimento multilivello si fa riferimento a tre livelli di apprendimento: apprendimento individuale (una percentuale del tempo del personale che viene dedicata a seguire le proprie passioni), apprendimento di gruppo (esperienze formative) e apprendimento organizzativo (strutturato tramite programmi di formazione aziendali). L'apprendimento inter-funzionale viene invece coltivato incoraggiando il personale ad accumulare esperienza in aree diverse dalla propria specializzazione.

Controllo sottile. I membri del team non vengono abbandonati durante il progetto di sviluppo; il management deve essere in grado di strutturare delle modalità di controllo del team che non siano rigide, in quanto bloccherebbero la creatività del team, ma che permettano allo stesso tempo di evitare instabilità, ambiguità e tensione.

Trasferimento dell'apprendimento. L'apprendimento non si limita alle modalità multilivello e inter-funzionale, ma deve avvenire anche tra i membri del team di sviluppo e gli altri membri dell'organizzazione. Questo può avvenire "per osmosi", ovvero mescolando i membri del team, o tramite la creazione di standard.

L'articolo di Takeuchi e Nonaka ha contribuito a rivoluzionare la pratica del Project Management e, come indicato in (Sutherland, 2012), rappresenta una delle tre pietre miliari che hanno contribuito alla formalizzazione del framework Scrum.

Come riportato all'interno dell'introduzione, Jeff Sutherland è uno dei due creatori di Scrum. All'inizio della sua carriera svolge undici anni di servizio militare come pilota dell'aviazione e successivamente si laurea alla University of Colorado School of Medicine, dove si appassiona di sviluppo di sistemi IT.

Nel 1993, Sutherland viene assunto come vicedirettore della sezione *Object Technology* della *Easel Corporation*, popolare società di sviluppo software, con l'obiettivo di realizzare in soli 6 mesi una linea di prodotti completamente nuova.

Il team è da subito consapevole che le vecchie modalità di gestione dei progetti di sviluppo sono inadeguate. Ciò li ha portati a realizzare una profonda analisi della letteratura e delle esperienze di centinaia di leader nello sviluppo di prodotti, alla ricerca di un nuovo metodo che consentisse loro di modificare il processo e adattarlo alla necessità di uno sviluppo rapido di applicazioni.

Come in seguito riportato, Sutherland è stato influenzato, oltre che da Nonaka e Takeuchi, anche dai principi del *Lean System* e dalle teorie sulla *Learning Organization* di *Peter Senge*.

“Our scalable, team-based all-at-once model was motivated by the Japanese approach to new product development. We were already using an iterative and incremental approach to building software. It was implemented in slices in which an entire piece of fully integrated functionality worked at the end of an iteration. What intrigued us was Hirotaka Takeuchi and Ikujiro Nonaka’s description of the team-building process for setting up and managing a SCRUM. The idea of building a self-empowered team where everyone had the global view of the product on a daily basis seemed to be the right one. The approach to managing the team which had been so successful at Honda, Canon, and Fujitsu also resonated with the systems thinking approach promoted by Professor Peter Senge at MIT.” (Sutherland, 2004)

Sutherland e il team gestiscono il progetto di sviluppo della nuova linea implementando il primo Scrum formale della storia, riuscendo a consegnare il prodotto nella data pianificata.

La consapevolezza raggiunta dal team con la prima implementazione del nuovo metodo, in accordo con (Sutherland, 2004), è la seguente:

- Scrum consente di rispettare le scadenze;
- Scrum consente di realizzare più funzionalità del previsto;
- Il metodo waterfall non risulta conveniente, per i seguenti motivi: non prevede, non rispetta le scadenze, produce meno funzionalità per sviluppatore per unità di tempo e genera una pessima soddisfazione nel cliente.

Nel 1995, Jeff Sutherland presenta il team Scrum di Easel a Ken Schwaber (co-autore di Scrum) che, quasi contemporaneamente, stava sperimentando la sua implementazione in Advanced Development Methods¹¹.

Sutherland e Schwaber raffinano e formalizzano i concetti appresi nel corso di anni di studi e applicazioni e, nel 1995, condividono alla conferenza OOPSLA il framework Scrum.

¹¹ Società di consulenza nello sviluppo software, fondata nel 1985 con sede in Massachusetts (USA).

1.2 Il Manifesto Agile per lo Sviluppo Software

A partire dagli anni '90 nascono diverse metodologie volte al superamento dell'approccio di sviluppo waterfall (i cui limiti sono descritti nell'introduzione dell'elaborato), tra cui Scrum, eXtreme Programming¹², DSDM e altri. I rappresentanti di tali metodologie decidono di riunirsi con l'obiettivo di individuare i punti in comune nel loro pensiero in quanto, fino a quel momento, non esisteva un modo coerente per descriverli (Agile Alliance, 2001).

Nasce quindi il *Manifesto Agile per lo Sviluppo Software*, un insieme di 4 valori e 12 principi definiti nel 2001 da 17 sviluppatori nello Snowbird Resort in Utah, USA. Viene di seguito riportato il contenuto del manifesto:

*“Stiamo scoprendo modi migliori di creare software,
sviluppendolo e aiutando gli altri a fare lo stesso.
Grazie a questa attività siamo arrivati a considerare importanti:
Gli individui e le interazioni più che i processi e gli strumenti
Il software funzionante più che la documentazione esaustiva
La collaborazione col cliente più che la negoziazione dei contratti
Rispondere al cambiamento più che seguire un piano
Ovvero, fermo restando il valore delle voci a destra,
consideriamo più importanti le voci a sinistra.”* (Agile Alliance, 2001)

I 12 principi sottostanti al Manifesto Agile sono i seguenti:

*“La nostra massima priorità è soddisfare il cliente
rilasciando software di valore, fin da subito
e in maniera continua.
Accogliamo i cambiamenti nei requisiti,
anche a stadi avanzati dello sviluppo.
I processi agili sfruttano il cambiamento
a favore del vantaggio competitivo del cliente.
Consegniamo frequentemente software funzionante,
con cadenza variabile da un paio di settimane a un paio di mesi,
preferendo i periodi brevi.
Committenti e sviluppatori devono lavorare insieme
quotidianamente per tutta la durata del progetto.*

¹² È una metodologia di sviluppo del software che enfatizza la scrittura di codice di qualità e la rapidità di risposta ai cambiamenti di requisiti.

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

*Una conversazione faccia a faccia
è il modo più efficiente e più efficace per comunicare
con il team ed all'interno del team.*

Il software funzionante è il principale metro di misura di progresso.

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

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

*La semplicità - l'arte di massimizzare la quantità
di lavoro non svolto - è essenziale.*

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

*A intervalli regolari il team riflette su come
diventare più efficace, dopodiché regola e adatta
il proprio comportamento di conseguenza.” (Agile Alliance, 2001)*

Dal momento della formalizzazione del manifesto all'interno dell'ambiente di sviluppo, e in seguito in quello organizzativo, si è diffuso il termine *agile*. Esso indica la capacità di creare e rispondere ai cambiamenti, un modo di affrontare e alla fine riuscire in un ambiente incerto e turbolento. In definitiva, agile è un *mindset*¹³ definito dai valori alla base del manifesto agile e dai 12 principi contenuti in esso. Questi valori forniscono una guida su come rispondere al cambiamento e su come affrontare l'incertezza (Agile Alliance, 2001).

Il manifesto Agile non fornisce degli step concreti per la realizzazione di prodotti, ma dà soltanto dei principi da seguire.

Scrum è una metodologia agile, in quanto rispecchia i principi contenuti nel manifesto Agile, ma i due termini non sono intercambiabili; per inquadrarne correttamente il rapporto viene riportata una spiegazione di Jeff Sutherland tratta da (Sutherland, 2020): “*Scrum e XP¹⁴ sono i genitori di Agile, che non è un framework o un'implementazione operativa. L'implementazione operativa dell'Agile in oltre l'80% dei team è oggi una variante di Scrum.*”

¹³ Mentalità, modo di pensare.

¹⁴ Abbreviazione di extreme programming.

1.3 Il Framework Scrum

“Scrum is an iterative, incremental framework for projects and product or application development.” (Sutherland, 2012)

“Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.” (Sutherland & Schwaber, 2017)

Come definito da Sutherland e Schwaber, Scrum non è un processo o una tecnica per costruire prodotti, ma piuttosto un framework, all'interno del quale è possibile utilizzare vari processi e tecniche, che consente di sviluppare prodotti nel dominio dei sistemi complessi¹⁵.

Le regole di Scrum prevedono 3 ruoli, 5 eventi e 3 artefatti. Implementare il framework in assenza di anche uno solo di questi elementi non porterebbe ad un'applicazione che possa essere chiamata Scrum (Sammicheli, 2018).

L'immagine sottostante [Figura 1] riassume graficamente l'intero funzionamento del framework:

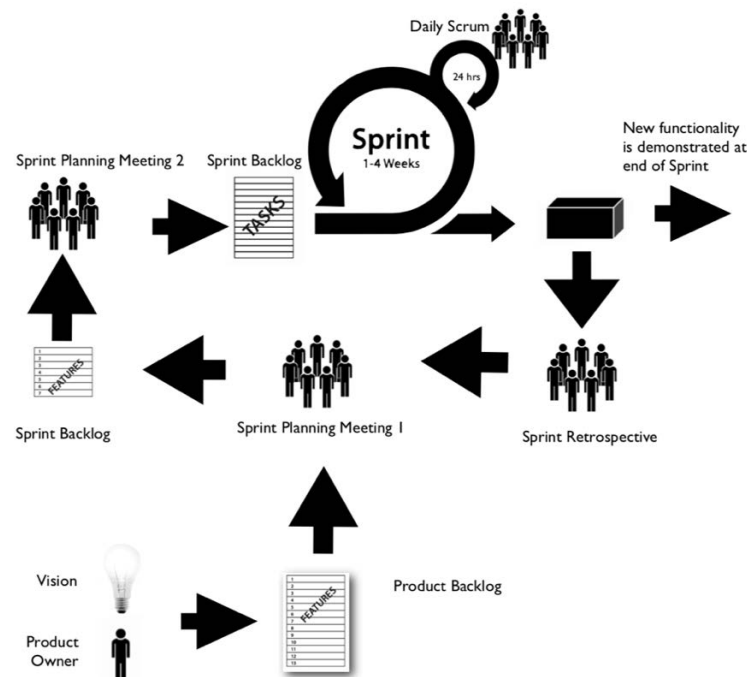


Figura 1 - Riassunto grafico del funzionamento di Scrum. Fonte: (Sutherland, The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012)

¹⁵ In relazione al modello Cynefin: “La situazione di un sistema Complesso è quella in cui non sappiamo cosa non sappiamo (Unknown Unknowns). La strategia di management associata a questo sistema è Probe - Sense - Respond, che prevede di effettuare un esperimento (probe), osservare i risultati (sense) e rispondere di conseguenza (respond), probabilmente con un nuovo esperimento in grado di ampliare la nostra conoscenza. In questo ambito l’approccio è iterativo ed incrementale e le pratiche sono dette Emergent Practices proprio perché emergono durante il processo di scoperta.” (Sutherland & Schwaber , La Guida a Scrum, 2016)

1.3.1 La teoria di Scrum

Scrum si basa sulla teoria del controllo empirico dei processi, o empirismo. L'empirismo è una corrente filosofica nata a metà del XVII secolo in Inghilterra. Secondo gli empiristi, la conoscenza umana deriva esclusivamente dai sensi o dall'esperienza, in contrapposizione con i razionalisti, secondo cui la filosofia dovrebbe essere condotta tramite l'introspezione e il ragionamento deduttivo a priori.¹⁶

Come ribadito nell'introduzione, in un contesto VUCA le organizzazioni non possono permettersi di irrigidire i propri processi e adottare metodologie di pianificazione che non considerano la possibilità di accadimento di eventi imprevedibili, o che non consentono di incorporare la conoscenza acquisita durante l'avanzamento del progetto. Per ottimizzare la prevedibilità e il controllo del rischio, Scrum utilizza un metodo iterativo e un approccio incrementale, in opposizione alla logica sequenziale dei metodi Stage-Gate e waterfall.

L'implementazione del controllo empirico dei processi si basa su 3 pilastri:

Trasparenza. Gli aspetti significativi del processo devono essere visibili ai responsabili del risultato finale (*out come*). In modo da condividere una visione comune su quello che è il lavoro che è stato svolto, risulta necessario formalizzare i diversi aspetti in uno standard (Sutherland, 2012).

Ispezione. I diversi attori coinvolti nell'utilizzo del framework devono ispezionare frequentemente gli artefatti¹⁷ di Scrum e monitorare l'avanzamento verso il raggiungimento degli obiettivi prefissati. L'ispezione consente di rilevare le eventuali deviazioni indesiderate.

Adattamento. Il rilevamento di deviazioni indesiderate deve permettere di adottare delle contromisure, ovvero adattare il processo o il materiale ad esso relativo nel più breve tempo possibile, in modo tale da minimizzare le deviazioni.

¹⁶ <https://it.wikipedia.org/wiki/Empirismo>

¹⁷ Verranno descritti in seguito.

1.3.2 I Ruoli di Scrum

In Scrum vi sono 3 ruoli fondamentali: Scrum Master, Product Owner e Team di Sviluppo. Queste figure assieme formano lo Scrum Team.

Scrum Master. È la figura responsabile del successo di Scrum; assume la veste di coach/insegnante affinché i principi e il funzionamento del framework siano compresi da tutti.

Viene definito un “*leader a servizio dello Scrum Team*”. Lo Scrum Master, in accordo con (Sutherland, 2012), ha l’obiettivo di massimizzare la generazione di valore del Team Scrum facendo capire ai membri esterni ad esso se le loro iterazioni siano a valore o meno. Scrum permette di rendere visibili alcuni impedimenti che si manifestano nel corso del progetto a Product Owner e team di sviluppo, di conseguenza il lavoro dello Scrum Master risulta fondamentale per eliminare tali impedimenti.

La figura dello Scrum Master può essere ricoperta da figure derivanti da svariate discipline e può essere condotta anche da un Project Manager, il quale deve però modificare il proprio mindset per vestire con successo tale ruolo. Lo Scrum Master deve infatti promuovere la cultura del problem solving nel team, utilizzando ad esempio l’approccio Socratico¹⁸, e facilitare il processo e l’auto-organizzazione piuttosto che semplicemente prendere una decisione ed assegnare l’attività al team.

Product Owner. Ha la responsabilità di massimizzare il ROI¹⁹ del progetto, il valore del prodotto e del lavoro svolto dal team di sviluppo (Sutherland, 2012). La figura del Product Owner rappresenta i bisogni degli *stakeholder*²⁰ del progetto ed è orientata alla loro soddisfazione. Nel caso si tratti di un progetto interno il Product Owner e il cliente coincidono. Nel caso di progetto esterno i clienti possono essere numerosi e con bisogni eterogeni; in questo caso il ruolo del Product Owner è simile al Product Manager, ma differisce da esso in quanto interagisce frequentemente e in modo attivo con il team di sviluppo, analizzando le priorità e i risultati.

In accordo con (Sutherland, 2012), il Product Owner è l’unica figura responsabile della gestione del Product Backlog (uno degli artefatti di Scrum,

¹⁸ Il metodo di porre domande che sono specificamente intese per aiutare gli altri ad imparare è detto *interrogatorio socratico*, in riferimento al Socrate dell’antica Grecia. Le domande socratiche prevedono un ascolto molto attento dell’altra persona affinché ciò aiuti a giudicare ed a formulare la propria domanda in modo utile, costruttivo e, si spera, non conflittuale.

¹⁹ *Return on Investment*, è un indicatore utilizzato per misurare l’utile/perdita, o il “valore” finanziario, di un progetto in relazione al suo costo. Calcolato come $[(\text{Valore} - \text{costo})/\text{costo}] * 100$.

²⁰ In economia lo *stakeholder* (in inglese letteralmente «titolare di una posta in gioco») o interessato è genericamente qualsiasi soggetto (o un gruppo) influente nei confronti di una iniziativa economica, una società o un qualsiasi altro progetto.

che verranno descritti in seguito), che include le seguenti attività: gli elementi del Backlog devono essere espressi in modo chiaro; gli elementi del Backlog devono essere ordinati in modo da permettere al team di raggiungere meglio gli obiettivi; il lavoro del team deve essere ottimizzato; gli elementi del Backlog devono essere compresi dal team di sviluppo.

Il ruolo del Product Owner è ispirato alla figura dello *Chief Engineer* di Toyota, il quale è responsabile di un'intera linea di prodotti.

Come evidenziato in (Deemer, Benefield, Larman, & Vodde, 2012), uno dei rischi della combinazione del ruolo dello Scrum Master con quello del Product Owner risiede nella creazione di un Product Owner micro-manager, il quale risulterebbe in contrasto con le caratteristiche dei team richiesti da Scrum e porterebbe ad una situazione conflittuale.

Team di Sviluppo. È responsabile della realizzazione del prodotto che viene definito dal Product Owner e di raggiungere gli obiettivi delle iterazioni. Le caratteristiche del team di sviluppo sono le seguenti:

- Le dimensioni del team devono oscillare da un minimo di 3 ad un massimo di 9. Team con un numero inferiore di 3 elementi presentano, molto probabilmente, delle carenze in termini di skills che comportano difficoltà a sviluppare un incremento di prodotto nel corso di un'iterazione. Team composti da più di 9 elementi comportano un effort eccessivo di coordinamento e generano complessità. In modo da aumentare la produttività e l'efficacia dei team, i membri dovrebbero essere dedicati completamente ad un singolo progetto di sviluppo; così facendo è possibile evitare gli effetti negativi del multitasking, come la suddivisione di attenzione e il cambio frequente di contesto (Deemer, Benefield, Larman, & Vodde, 2012).
- Autonomia: il team è auto-organizzato e auto-gestito, libero di decidere come svolgere il proprio lavoro e autorizzato a mettere in pratica quelle decisioni. Nessuno, nemmeno lo Scrum Master, può dire al team come portare avanti lo sviluppo.
- Cross-funzionale: possiede tutte le competenze per portare a termine lo sviluppo del prodotto. I singoli membri possiedono delle competenze specialistiche o aree di focus, ma è il team di sviluppo nel suo complesso ad avere la responsabilità finale. Non sono inoltre permessi dei sotto-team dedicati ad alcuni aspetti specialistici dello sviluppo (Sutherland, 2012).

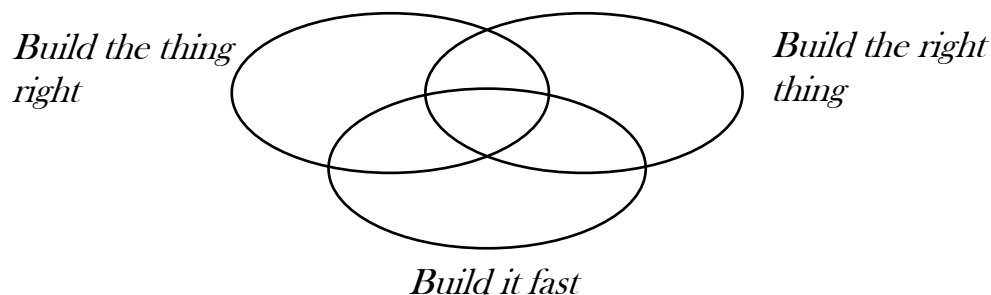
Come evidenziato in (Sammicheli, 2018): “Può succedere che persone con skills specialistiche, impiegate in maniera non continuativa durante

il ciclo di sviluppo, si trovino sottoutilizzate in alcune fasi. Per evitare che ciò accada, e per permettere l'autorganizzazione del team, Scrum incoraggia lo sviluppo, in tutti i componenti del team, di competenze organizzate in forma di T (T-Shaped Skills). La T rappresenta in questo caso la forma di una matrice delle competenze in cui ciascun componente del team ha una competenza di base in ciascun ambito richiesto dallo sviluppo del prodotto, e ha allo stesso tempo un livello di conoscenza avanzato in uno dei settori. Tuttavia, è necessario creare le condizioni affinché le persone sviluppino competenze pi-shaped, ovvero a forma di p greco, con una competenza primaria molto forte, una competenza secondaria discreta e una competenza base su tutti gli altri aspetti.” In questo modo i membri del team sono incoraggiati ad apprendere anche altre specialità.

- Non ci sono titoli specialistici, ma ogni membro del team è solamente un componente del team. Il possesso di un titolo specialistico comporta che ciascun membro del team si focalizzi solamente sulle attività che gli competono, portando ad un accumulo di conoscenza specifica e di potere. Eliminare la specializzazione, espressa in numero di ruoli e titoli, porta ad un miglioramento della comunicazione del team e, di conseguenza, ad un incremento della velocità (Sutherland, 2015).

In accordo con (Sammicheli, 2018), nello sviluppo di un prodotto, di qualsiasi tipo, vi sono 3 obiettivi che si vorrebbero raggiungere:

- *Build the right thing*: costruire il prodotto con funzionalità tali da permettere la soddisfazione dei bisogni dei clienti.
- *Build the thing right*: costruire il prodotto bene dal punto di vista ingegneristico.
- *Build it fast*: costruire il prodotto con un basso time to market.²¹



Progetti di sviluppo con performance disallineate dal centro della figura comportano da un lato clienti insoddisfatti e dall'altro bassa redditività per l'azienda.

La struttura del framework Scrum permette ai progetti di sviluppo di mantenersi nella posizione di performance ideale, ovvero al centro, tramite la suddivisione delle responsabilità tra i diversi ruoli: il Product Owner deve occuparsi di definire le funzionalità che servono veramente: *“Build the right thing”*; il Team di Sviluppo si concentra a realizzare in modo tecnicamente valido le funzionalità richieste: *“Build the thing right”*; infine, lo Scrum Master si concentra sul processo, facilitando il lavoro del team ed eliminando gli sprechi: *“Build it fast”* (Sammicheli, 2018).

²¹ Il *time to market* (o TTM) è un'espressione anglofona che indica il tempo che intercorre dall'ideazione di un prodotto alla sua effettiva commercializzazione.

1.3.3 Lo Sprint

Lo *Sprint* è il cuore pulsante di Scrum e può essere considerato come un progetto con un orizzonte temporale più breve rispetto all'intero progetto di sviluppo, all'interno del quale si svolgono tutti gli eventi previsti da Scrum.

L'obiettivo di lavoro di uno Sprint è raggiungere lo *Sprint Goal*, ovvero incrementi di funzionalità del prodotto potenzialmente rilasciabili. Lo Sprint Goal può essere qualunque tipo di coerenza che faccia sì che il Team di Sviluppo possieda flessibilità, in rapporto alla funzionalità implementata, e lavori insieme, piuttosto che secondo iniziative individuali (Sutherland & Schwaber, 2017). In questo modo è possibile formalizzare il "perché" dello Sprint, in modo tale che il team possa definire in autonomia il "come".

Lo Sprint segue una logica *time-boxed*, che consiste in una finestra temporale di lavoro con durata definita, fino ad un massimo di 4 settimane, che si interrompe al termine prestabilito anche se il team di sviluppo non ha portato a termine il lavoro, per poi dare avvio ad una nuova finestra temporale. La durata dello Sprint non è mai prolungabile. Eseguire Sprint in successione permette di creare un flusso di lavoro cadenzato all'interno del processo di sviluppo e consente di implementare il principio del miglioramento continuo: il team elimina gli sprechi intrinseci dello sviluppo con l'obiettivo di massimizzare l'output generato ad ogni iterazione.

Come definito in (Sutherland & Schwaber, 2016): "Lo Sprint permette di applicare i principi di ispezione ed adattamento del controllo empirico dei processi ad ogni finestra temporale, riducendo inoltre il rischio ad una singola iterazione." Quando il team fissa l'obiettivo per lo Sprint, questo non deve essere modificato fino allo Sprint successivo; così facendo il team si sente protetto rispetto ai cambiamenti inaspettati che possono sorgere. In particolare, vi sono due aspetti positivi che si generano: in primis, il team si focalizza fin da subito sull'obiettivo da raggiungere, conscio del fatto che non vi saranno delle interferenze che lo andranno a modificare. In secondo luogo, impedire le modifiche allo Sprint Goal nell'arco della durata dello Sprint implica una maggior focalizzazione del Product Owner su quelle che sono veramente le priorità dello sviluppo; allo stesso tempo però la presa in carico dei cambiamenti viene rimandata solamente della durata di uno Sprint. In questo modo è possibile superare lo stigma attorno al cambiamento (cambiamento di direzione, cambiamento dei requisiti, o semplicemente cambiamento di opinione) (Deemer, Benefield, Larman, & Vodde, 2012).

Uno Sprint può essere cancellato prima della scadenza prestabilita nel caso in cui lo Sprint Goal diventi obsoleto e il Product Owner è l'unico responsabile di tale decisione.

1.3.4 Gli Artefatti di Scrum

Gli artefatti di Scrum rappresentano il lavoro e il valore in diversi modi utili, al fine di fornire trasparenza e opportunità di ispezione e adattamento. Gli artefatti definiti da Scrum sono progettati per massimizzare la trasparenza delle informazioni chiave, in modo che ognuno abbia lo stesso livello di comprensione (Sutherland, 2012).

Backlog. Vi sono due tipologie di Backlog: il Product Backlog e lo Sprint Backlog.

Il *Product Backlog* rappresenta la fotografia in un preciso istante della conoscenza del team riguardo a quello che può essere sviluppato da quel momento in poi, in ordine di priorità; è in continua evoluzione e unico per ciascun prodotto che viene sviluppato. Contiene una serie di voci, principalmente nuove funzionalità per i clienti, ma anche obiettivi di miglioramento ingegneristico, lavori di ricerca o esplorazione, requisiti di prestazione o sicurezza e difetti noti (Sutherland, 2012). Il Product Backlog è di esclusiva gestione del Product Owner. In accordo con (Deemer, Benefield, Larman, & Vodde, 2012), un buon Product Backlog dovrebbe essere DEEP:

- *Dettagliato*: le voci che si trovano nella parte alta del Backlog, ovvero quelle che hanno una priorità maggiore, devono essere più affinate e dettagliate rispetto a quelle che si trovano nella parte bassa, in quanto saranno prese in considerazione prima dal team di sviluppo.
- *Stimato* (Estimated): le voci devono essere stimate dal team in base allo sforzo per portarle a termine e, in alcuni casi, anche in base al rischio. La stima deve essere aggiornata ad ogni iterazione di Scrum, in modo da tenere in considerazione la nuova conoscenza acquisita e i cambiamenti dell'ambiente. Il Product Owner e gli altri soggetti aziendali coinvolti forniscono al team le informazioni necessarie alla stima. Scrum non impone la forma della stima nel Product Backlog, ma è comune utilizzare stime relative espresse come "punti" piuttosto che unità di sforzo assolute come le settimane-persona (Sutherland, 2012). Sutherland in (Sutherland, 2015) suggerisce di utilizzare la sequenza di Fibonacci²², in quanto i numeri che la compongono sono abbastanza distanziati da permettere di cogliere immediatamente la differenza di carico di lavoro.
- *Emergente*: il Product Backlog viene aggiornato ad ogni iterazione: le voci possono essere create, eliminate, modificare o ri-prioritizzate, con

²² In matematica indica una successione di numeri interi in cui ciascun numero è la somma dei due precedenti, eccetto i primi due che sono, per definizione $F_0=0$ e $F_1=1$.

l'obiettivo di rispondere in modo tempestivo all'apprendimento e alla variabilità del processo. I cambiamenti delle esigenze del cliente, nuove idee o intuizioni, le mosse della concorrenza o ostacoli tecnici che si presentano si riflettono nella modifica e nell'aggiornamento del Product Backlog (Deemer, Benefield, Larman, & Vodde, 2012).

- **Prioritizzato:** le voci che compongono la parte superiore del Backlog devono essere ordinate da 1 a N in base al valore di business e al rischio [Figura 2]. In questo modo il completamento delle voci ad alta priorità fornisce rispettivamente un'alta redditività a basso costo e/o consente di affrontare i rischi elevati prima che questi si manifestino (Deemer, Benefield, Larman, & Vodde, 2012). Per stimare la priorità delle voci del Backlog si possono utilizzare diverse tecniche, come ad esempio i value-point²³.

Item	Details (wiki URL)	Priority	Estimate of Value	Initial Estimate of Effort
As a buyer, I want to place a book in a shopping cart (see UI sketches on wiki page)	...	1	7	5
As a buyer, I want to remove a book in a shopping cart	...	2	6	2
Improve transaction processing performance (see target performance metrics on wiki)	...	3	6	13
Investigate solutions for speeding up credit card validation (see target performance metrics on wiki)	...	4	6	20
Upgrade all servers to Apache 2.2.3	...	5	5	13

Figura 2 - Esempio di Product Backlog. Fonte: (Sutherland, 2012)

Lo *Sprint Backlog* rappresenta il corrispettivo del Product Backlog per il singolo Sprint ed è composto dall'insieme delle voci che il team di sviluppo ritiene di essere in grado di portare a termine nel successivo Sprint e i task necessari per completarle [Figura 3]. Le voci da portare a termine vengono pianificate all'inizio dello Sprint dal team di sviluppo, in un evento dedicato, e possono essere modificate nel corso dello Sprint se il team lo ritiene necessario. Lo Sprint Backlog è di gestione esclusiva del Team di Sviluppo.

Product Backlog Item	Sprint Task	Volunteer	Initial Estimate of Effort
As a buyer, I want to place a book in a shopping cart	modify database		5
	create webpage (UI)		8
	create webpage (Javascript logic)		13
	write automated acceptance tests		13
	update buyer help webpage		3
	...		
Improve transaction processing performance	merge DCP code and complete layer-level tests		5
	complete machine order for pRank		8
	change DCP and reader to use pRank http API		13

Figura 3 - Esempio Sprint Backlog. Fonte: (Sutherland, 2012)

²³ Punteggio che sintetizza una "stima approssimativa" di fattori tra cui l'aumento dei ricavi, riduzione dei costi, le preferenze degli stakeholder, la differenziazione del mercato, e così via.

Gli elementi presenti all'interno del Backlog possono essere descritti secondo diverse modalità e, contrariamente al pensiero diffuso, il Backlog non contiene *User Stories*, ma voci; tali voci possono essere espresse come User Stories, o utilizzando altri approcci per esprimere i requisiti che il gruppo trova utili. Le voci possono essere ad esempio funzionalità del prodotto, obiettivi interni, obiettivi di miglioramento o altro. Qualunque sia l'approccio utilizzato, la maggior parte delle voci dovrebbero concentrarsi sul fornire valore ai clienti (Deemer, Benefield, Larman, & Vodde, 2012).

Le User Stories rappresentano la forma più diffusa per descrivere una voce presente all'interno del Backlog e consistono in brevi frasi che contengono tre aspetti fondamentali per ciascun elemento: Chi, Cosa e Perché.

Uno degli schemi più utilizzati è il seguente:

“In veste di <Chi>, voglio <Cosa: Attività> in modo da <Perché: Beneficio>”.

È fondamentale sottolineare che tutti devono avere chiaro e comprendere quando un elemento del Backlog è considerato “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 intende per lavoro completo, in modo da poter implementare i principi di trasparenza ed ispezione sull'incremento di prodotto (Sutherland & Schwaber, 2016).

Come indicato in (Smith, 2007), raramente i requisiti rimangono stabili nel corso del progetto e di conseguenza, contrariamente a quanto avviene nell'approccio tradizionale, provare a specificare completamente il prodotto prima di iniziare la progettazione provoca un allungamento significativo dei tempi del progetto. L'obiettivo deve essere quello di fornire valore al cliente e non di proteggere il prodotto contro qualsiasi cosa un concorrente possa aggiungere.

Scrum, e in particolare lo strumento del Backlog, consente di tenere in considerazione cambiamenti nei requisiti e adattare il prodotto per fornire il massimo valore al cliente [Figura 4].

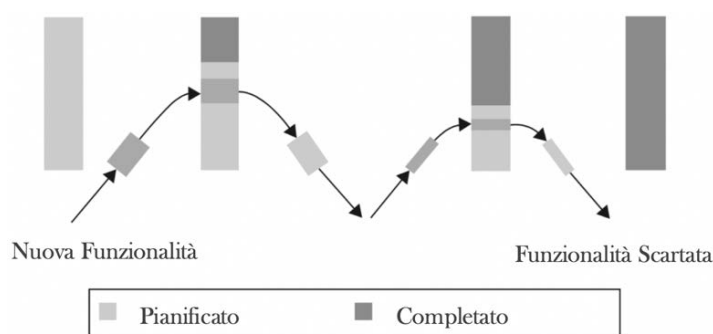


Figura 4 - Cambiamento di requisiti nelle varie iterazioni. Fonte: (Smith, 2007)

Incremento. In (Sutherland & Schwaber, 2017) l'incremento di prodotto viene definito come *“la somma di tutti gli elementi del Product Backlog completati durante uno Sprint e del valore degli incrementi di tutti gli Sprint precedenti”*. L'incremento di prodotto al termine dello Sprint deve risultare “fatto”, ovvero deve essere coerente con lo standard che è stato assegnato dal team. L'incremento consente al team di accumulare conoscenza tramite l'ispezione, in modo da supportare il controllo empirico, e di avanzare di un passo verso il raggiungimento della visione.

Per incrementare la trasparenza riguardo l'avanzamento del lavoro del team nello Sprint e il raggiungimento dello Sprint Goal, e quindi creare occasioni di ispezione e adattamento, vengono utilizzati degli strumenti di supporto:

Scrum Board. È uno strumento che consente di rendere trasparente il lavoro del team di sviluppo tramite l'utilizzo di una lavagna, o un cartellone, in cui post-it colorati si muovono lungo 3 colonne principali [Figura 5]:

- *Da fare:* contiene le voci dello Sprint Backlog che devono essere ancora iniziate;
- *In corso:* contiene le voci che i membri del team hanno iniziato a sviluppare;
- *Fatte:* contiene le voci che i membri del team hanno concluso, coerentemente con la “definizione di fatto” concordata.

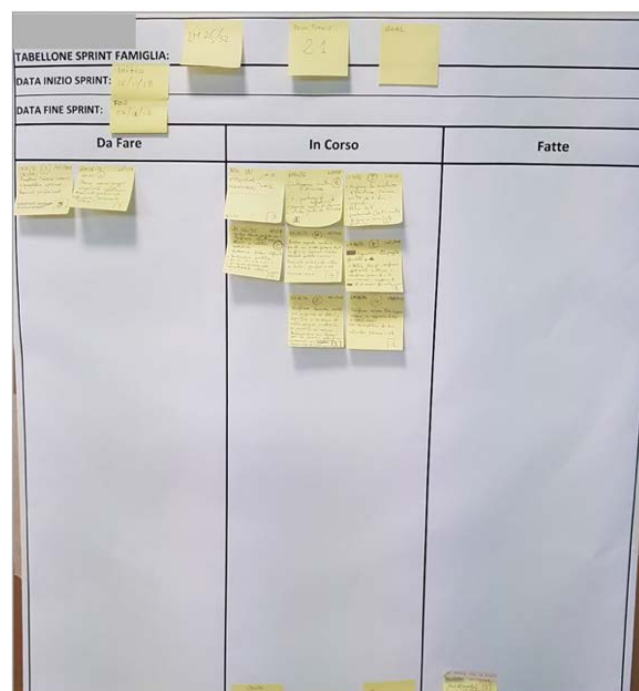


Figura 5 - Esempio Scrum Board. Fonte: auxiell

In accordo con (Sutherland, 2015), l'utilizzo della Scrum Board è fondamentale in quanto consente di capire cosa è stato fatto e cosa manca da fare; permette al team di auto-regolarsi e di auto-organizzarsi per risolvere problemi che diventano evidenti solamente quando tutto è trasparente, come nel caso in cui una voce rimanga ferma nella colonna “In corso” per troppo tempo. Consente inoltre, a chiunque entri nella stanza dove il team lavora, di capire con una semplice occhiata l'andamento del team.

Sprint Burn Chart. È uno strumento che consente di rendere trasparente la velocità del team di sviluppo nei diversi Sprint.

Lo *Sprint Burn-down Chart* [Figura 6] consente di visualizzare a tutti il lavoro rimanente al team per raggiungere l'obiettivo. Viene utilizzato un grafico in cui nell'asse orizzontale è indicato il numero di giorni dello Sprint mentre nell'asse verticale è indicato il punteggio cumulato delle voci che sono state selezionate dal team. Ogni giorno il grafico viene aggiornato, sottraendo al dato di partenza il punteggio delle storie che sono state completate il giorno precedente. Idealmente la curva dovrebbe avere un andamento decrescente e raggiungere il valore zero nell'ultimo giorno di lavoro dello Sprint.

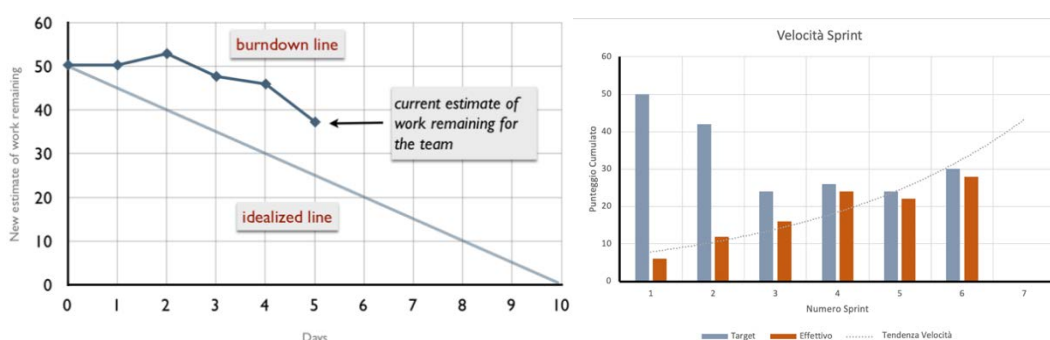


Figura 6 – A sinistra, esempio di Sprint Burn-down Chart. Fonte: (Sutherland, The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012)

Figura 7 – A destra, esempio Sprint Burn-up Chart

Lo *Sprint Burn-up Chart* [Figura 7] monitora la velocità del team mappando il lavoro che viene completato nei diversi Sprint, ovvero la curva cumulata dei punteggi delle voci “fatte”. In questo caso, nell'asse delle ascisse sono indicati i diversi Sprint e la curva dovrebbe assumere un andamento crescente, in quanto l'obiettivo del Team Scrum è quello di aumentare la velocità di sviluppo man mano che il progetto avanza, in modo da “Fare il Doppio in Metà Tempo”²⁴.

²⁴ Titolo del Libro di Jeff Sutherland, co-inventore di Scrum.

1.3.5 Gli Eventi di Scrum

Gli eventi prescritti hanno l'obiettivo di creare regolarità e ridurre al minimo la necessità di incontri non definiti in Scrum.

Oltre allo stesso Sprint, che è un contenitore di tutti gli altri eventi, ogni evento in Scrum è un'occasione formale per ispezionare e adattare qualcosa. Questi eventi sono specificamente progettati per consentire trasparenza critica e ispezione. La mancata inclusione di uno qualsiasi di questi eventi comporta una riduzione della trasparenza ed è un'occasione mancata per praticare l'ispezione e l'adattamento, ovvero per mettere in pratica il controllo empirico dei processi (Sutherland & Schwaber, 2016).

Sprint Planning. Si svolge all'inizio di ogni Sprint e ha come obiettivo la pianificazione dello Sprint stesso. È suddiviso in due parti:

Lo *Sprint Planning Parte Prima* coinvolge l'intero Team Scrum e ha come obiettivo definire "cosa dovrà essere fatto nello Sprint": il Product Owner e il Team discutono e analizzano le voci ad alta priorità del Product Backlog per il quale il Product Owner è interessato all'implementazione nello Sprint che sta per iniziare (Deemer, Benefield, Larman, & Vodde, 2012). Il Product Owner propone al Team lo Sprint Goal, ovvero le funzionalità potenzialmente rilasciabili al termine dello Sprint, e le voci del Product Backlog che possono essere prese in considerazione per raggiungere l'obiettivo. Il Team di Sviluppo discute la proposta dello Sprint Goal e concorda le storie da analizzare.

La prima parte è focalizzata sulla comprensione di *quali* funzionalità il Product Owner richiede e *perché* sono necessarie (Deemer, Benefield, Larman, & Vodde, 2012).

Lo *Sprint Planning Seconda Parte* ha come obiettivo definire "come portare a termine il lavoro": il Team di Sviluppo definisce le voci del Backlog che possono essere portate a termine nel corso dello Sprint sulla base delle proprie analisi, entrando a far parte dello Sprint Backlog, e i task necessari al loro completamento.

La seconda parte dello Sprint Planning non prevede una partecipazione attiva del Product Owner, che deve comunque essere reperibile nel caso di perplessità del team, in quanto il Team pianificherà lo sviluppo di voci a più alta priorità con l'obiettivo di raggiungere lo Sprint Goal concordato. Il Team ha la possibilità di fare pressioni per voci che si trovano più in basso nella lista; questo di solito accade quando il Team e il Product Owner si rendono conto che qualcosa di priorità inferiore si adatta facilmente e opportunamente con le voci ad alta priorità.

Scrum non definisce come pianificare le attività nella seconda parte; il punteggio delle voci che il Team si prende in carico può essere stimato sulla base della velocità degli Sprint precedenti oppure sulla base della capacità disponibile e la stima dell'effort richiesto dai task (espresso in ore).

Lo Sprint Planning, come gli altri eventi di Scrum, è time-boxed: la durata è di un'ora per ciascuna parte per ogni settimana di Sprint (quindi otto ore al massimo, nel caso di uno Sprint di quattro settimane).

È importante sottolineare che il Product Owner guida e indirizza il Team verso le funzionalità richieste e le voci a maggior priorità, ma è il Team che in autonomia decide cosa fare e come portare a termine il lavoro nel corso dello Sprint.

Backlog Refinement. Una determinata percentuale del tempo del Team (circa il dieci per cento della durata dello Sprint) deve essere dedicata alla raffinazione del Backlog, che comprende l'analisi dettagliata dei requisiti, la suddivisione di voci di grandi dimensioni in voci più piccole, la stima di nuove voci, e la nuova stima di quelle esistenti (Deemer, Benefield, Larman, & Vodde, 2012).

Questa attività non ha lo scopo di affinare le voci selezionate per lo Sprint in corso, ma è dedicata alle voci da realizzare in futuro, molto probabilmente nei successivi due Sprint. La raffinazione del Backlog (denominata anche *grooming*) rappresenta quindi un driver fondamentale per facilitare la riunione di Sprint Planning, in quanto il Team Scrum può pianificare avendo a disposizione un'analisi ed una stima accurata delle voci.

All'interno della teoria di Scrum non viene definita una metodologia con la quale effettuare la raffinazione del Backlog; una delle modalità con la quale si può svolgere l'evento è un workshop di raffinazione, che viene tipicamente svolto a metà o alla fine dello Sprint. Il focus dell'evento deve essere quello di permettere al Team, Product Owner e altri soggetti interessati di concentrarsi senza interruzioni (Deemer, Benefield, Larman, & Vodde, 2012).

Daily Scrum. È una breve riunione, della durata massima di quindici minuti, che viene svolta ogni giorno alla stessa ora, in cui i membri del Team sono invitati a rimanere in piedi in modo da contenerne la durata. L'obiettivo è riportare i progressi verso lo Sprint Goal e sincronizzare l'avanzamento del lavoro, tramite il monitoraggio delle voci completate nella Sprint Board e l'aggiornamento dello Sprint Burndown Chart.

Nel Daily Scrum ogni membro del team, uno alla volta, riferisce su tre cose agli altri membri del Team: (1) Che cosa è stato fatto dopo l'ultima riunione; (2) Che cosa sarà fatto prima della prossima riunione; (3) Quali ostacoli ci sono da affrontare (Deemer, Benefield, Larman, & Vodde, 2012).

Il ruolo dello Scrum Master è quello di far rispettare le regole e di rimuovere eventuali ostacoli che si sono manifestati al team, mentre il Product Owner (la cui presenza non è obbligatoria) può fornire al team eventuali brevi chiarimenti riguardo al Backlog. Nel caso in cui siano necessari degli approfondimenti, vengono discussi in incontri di *follow-up*, in cui alcuni o tutti i membri del Team si adeguano alle informazioni che hanno sentito nel Daily Scrum (Deemer, Benefield, Larman, & Vodde, 2012).

Il Daily Scrum è un evento fondamentale per raggiungere l'ispezione e l'adattamento; gli obiettivi che tale riunione si prefissa di raggiungere sono il miglioramento della comunicazione, la rimozione di altri incontri non pianificati, l'identificazione e l'eliminazione degli ostacoli, la promozione di un rapido processo decisionale e il miglioramento della conoscenza del progetto da parte del team (Sutherland & Schwaber, 2016).

Sprint Review. Si svolge al termine di ogni Sprint con l'obiettivo di ispezionare l'incremento di prodotto e adattare, se necessario, il Product Backlog. La riunione ha la durata di un'ora per ogni settimana di Sprint e coinvolge lo Scrum Team e altri stakeholder invitati dal Product Owner.

Il Team di Sviluppo presenta il lavoro che è stato fatto nel corso dello Sprint precedente; inoltre, interagisce con i partecipanti riguardo agli aspetti positivi dell'iterazione che è terminata, i problemi riscontrati e come sono stati risolti. Il Product Owner ha il compito di identificare ciò che è stato "fatto", ovvero le voci dello Sprint Backlog concluse, e ciò che non è stato "fatto", per poi aggiornare il Product Backlog e fare una stima della data di consegna delle attività rimanenti. Si tratta di un evento informale in cui non ci si limita a "mostrare" l'incremento di prodotto, ma vi è una collaborazione tra il Team Scrum, ed eventuali stakeholder, con l'obiettivo di ricavarne feedback, consigli, maggior comprensione sulla situazione del mercato. Il raccoglimento di una maggior conoscenza genera in output una rivisitazione del Product Backlog, in cui sono resi evidenti gli elementi che probabilmente verranno presi in carico nel prossimo Sprint (Sutherland & Schwaber, 2016).

Sprint Retrospective. Si svolge successivamente allo Sprint Review, e prima dell'inizio del nuovo Sprint, con l'obiettivo di ispezionare e adattare il Team stesso. Mentre lo Sprint Review è un evento dedicato all'ispezione e adattamento con riferimento al prodotto, lo Sprint Retrospective è dedicato al processo e all'ambiente.

Lo scopo dello Sprint Retrospective è di:

- Esaminare come l'ultimo Sprint sia andato in merito a persone, relazioni, processi e strumenti;
- Identificare e ordinare gli elementi principali che sono andati bene e le migliorie potenziali;
- Creare un piano per attuare i miglioramenti al modo di lavorare dello Scrum Team. Tale piano di miglioramento può essere inserito come una voce all'interno del successivo Sprint Backlog.

Ha una durata di un'ora per ogni settimana di Sprint e coinvolge l'intero Scrum Team. Il ruolo dello Scrum Master è fare in modo che l'incontro sia proficuo e rappresenta l'elemento neutro del team in relazione alla sua responsabilità sul processo di Scrum (Sutherland & Schwaber, 2017).

Capitolo 2.

Scrum nello sviluppo Hardware

“Negli ultimi 33 anni, mi sono guardato allo specchio ogni mattina e mi sono chiesto: “Se oggi fosse l’ultimo giorno della mia vita, vorrei fare ciò che sto per fare oggi?” E ogni volta che la risposta è stata “No” per troppi giorni di fila, ho capito che bisognava cambiare qualcosa.”

Steve Jobs

L’obiettivo di questo capitolo è fornire un’analisi di letteratura sui diversi approcci che sono nati per favorire l’implementazione del framework Scrum nello sviluppo di prodotti tangibili e le sfide che si manifestano nella sua applicazione in tale contesto.

2.1 Le sfide dello sviluppo hardware con Scrum

Come illustrato all'interno del primo capitolo, Scrum nasce come framework di riferimento per lo sviluppo di software, ma come indicato dai creatori in (Sutherland & Schwaber, 2017) può essere utilizzato in svariati ambiti, tra cui lo sviluppo di prodotti tangibili.

Nella letteratura di Scrum applicato allo sviluppo hardware si fa riferimento a diverse sfide e opportunità su cui è necessario concentrarsi per l'implementazione del framework in tale contesto.

In particolare, in (Ullman, 2019) sono elencate e descritte 12 sfide, di cui in seguito sono riportate le più significative:

Modularità. La modularità di prodotto è ampiamente sfruttata all'interno dei prodotti software, permettendo di sviluppare i diversi moduli in maniera indipendente e riducendo quindi le implicazioni dei cambiamenti. Spesso all'interno dei prodotti tangibili il confine tra funzione e forma viene sorpassato, portando ad una più stretta interdipendenza nei cambiamenti tra diversi moduli. Questo effetto può essere controllato suddividendo il prodotto in sistemi e sottosistemi progettati attorno ad interfacce stabili e conosciute; questo comporta una definizione delle interfacce tra i diversi moduli nella fase iniziale del processo di sviluppo con l'obiettivo di renderle il più semplici e stabili possibile (Ullman, 2019).

Durata limitata degli Sprint. L'obiettivo ideale di ciascuno Sprint è quello di rilasciare un "incremento di funzionalità di prodotto potenzialmente rilasciabile", il quale rappresenta un traguardo sia nello sviluppo software che nello sviluppo hardware. I prodotti software sono la combinazione nel tempo di righe di codice, ciascuna delle quali rappresenta una sezione funzionale. Per quanto riguarda i prodotti hardware spesso non è possibile seguire la stessa logica di sviluppo in quanto, come riportato nella sezione precedente, è più comunemente costruito con componenti e assiemi che presentano caratteristiche e funzioni specifiche. Per tale motivo, l'obiettivo di uno Sprint nel contesto dello sviluppo hardware dovrebbe focalizzarsi sulla progettazione di una parte del prodotto, come un componente o un assieme, e non su una determinata funzione. Per far ciò è necessario che il team sia in grado di definire dei criteri di accettazione delle storie che risultino realistici e raggiungibili all'interno della durata di un singolo Sprint.

Aggiunta di nuove funzionalità. I prodotti software vengono realizzati attraverso rilasci multipli di versioni di prodotto, ciascuna delle quali modifica la precedente con l'aggiunta/eliminazione di funzionalità. Tale modus operandi, denominato *creep di funzionalità*²⁵, può risultare non raggiungibile nello sviluppo hardware a causa della più stretta interazione tra

²⁵ Aggiunta di nuove funzionalità durante il processo di progettazione. (Ullman, 2019)

forma e funzionalità e può provocare una perdita di controllo sui costi e sulle date di consegna del progetto.

Miglioramento e Ri-lavorazione. Se l'utilizzo di un particolare algoritmo nell'esecuzione di una funzione del software si dimostra inefficiente, non è costoso riscriverlo e migliorarlo, in quanto viene investito solamente il tempo necessario alla ri-scrittura. Al contrario, una ri-progettazione delle caratteristiche e funzioni di un prodotto fisico è solitamente più costosa (salari, attrezzature, inventario) e difficile. I miglioramenti nell'hardware sono garantiti da un'attenta definizione dei moduli e delle interfacce del prodotto (Ullman, 2019).

Specializzazione. Nel software, una persona in grado di codificare una determinata funzione è, molto probabilmente, in grado di codificarne altre. Nell'hardware, le figure coinvolte non hanno un'esperienza universale tra le diverse funzioni. Una persona che progetta un motore a combustione probabilmente non ha esperienza nella progettazione di cuscinetti auto-lubrificanti o componenti elettrici. Questo si riflette in una maggior cura nell'assegnazione dei compiti al team di sviluppo nel corso dello Sprint, che può avvenire con cadenza settimanale nello Sprint Planning; nel software invece i task possono essere assegnati giornalmente ai membri del team.

Dimostrazione delle funzioni. Contrariamente a quanto accade nello sviluppo software, dove i codici possono essere rapidamente scritti e compilati, nell'hardware la progettazione e la costruzione possono richiedere un tempo maggiore. Tecnologie quali Additive Manufacturing (es. stampa 3D), realtà virtuale (VR) e realtà aumentata (AR) sono di supporto a ridurre il tempo necessario a dimostrare le funzioni nell'hardware.

Test. Il test è uno sforzo per ridurre il rischio trovando ed eliminando in modo proattivo i problemi che possono influire sull'usabilità o l'affidabilità di un prodotto. I test software di solito includono migliaia di casistiche di errore, quindi i programmatori progettano un codice specializzato per sradicare potenziali problemi. Nell'hardware sono richieste meno casistiche di test, ma con attrezzature più specializzate e costose. L'apparecchiatura di collaudo può essere progettata e costruita come parte del progetto. Indipendentemente dal fatto che si stia progettando software o hardware, Scrum incoraggia lo sviluppo test-driven (TDD). TDD impone la definizione dei test di accettazione durante la definizione di ciascun compito. In altre parole, descrivere ciò che è necessario per dimostrare che un'attività viene eseguita, prima che l'attività venga iniziata (Ullman, 2019).

Sviluppo di sistemi ausiliari. Lo Sviluppo di prodotti tangibili non si limita alla progettazione dei disegni tecnici o analisi ma richiede la progettazione dell'intera catena del valore, ovvero la progettazione del processo di produzione, il processo di assemblaggio, le apparecchiature di prova, le procedure di controllo qualità/certificazione, la catena di fornitura e tutta la documentazione di supporto, oltre che il processo di vendita/post-vendita e la gestione della logistica interna ed esterna. Per il software, la scrittura del codice è il processo di produzione che genera il prodotto, non esiste una catena di approvvigionamento e la documentazione è generalmente più semplice rispetto ai sistemi hardware. Un approccio adottato per gestire la complessità è quello di portare tutti i problemi all'interno del team e considerare tutte le figure necessarie alla gestione dell'intera catena del valore.

2.2 Extreme Manufacturing

Extreme Manufacturing è l'approccio ideato da Joe Justice²⁶ per gestire iterazioni settimanali legate a prodotti tangibili, abilitando all'uso di Scrum anche nell'hardware (Sammicheli, 2018). Questo metodo ha permesso a Justice e a tutti gli altri membri del team Wikispeed di realizzare con un budget quasi nullo e un tempo limitato un'automobile immatricolabile con consumi inferiori a 100 MPGe²⁷.

Lo sviluppo di tale approccio nasce dalla necessità di trovare un metodo che permetta di generare in modo regolare un feedback di apprendimento in ciascuno Sprint di Scrum, anche nel caso in cui si stiano sviluppando dei prodotti tangibili tecnologicamente complessi. Nello sviluppo software si utilizzano delle pratiche ingegneristiche denominate *eXtreme Programming*, da cui eXtreme Manufacturing trae ispirazione. Tale approccio è caratterizzato da un insieme di principi e di pattern che aiutano a creare e adattare i prodotti velocemente e integrare le modifiche rapidamente nei prodotti esistenti. I principi alla base dell'eXtreme Manufacturing sono i seguenti (Sammicheli, 2018):

Ottimizzazione per il cambiamento. Il principio dell'ottimizzazione per il cambiamento prevede di analizzare in anticipo il modo in cui vengono realizzati i componenti che prevediamo di far evolvere più velocemente all'interno dei prodotti, in modo da tenere in considerazione già in fase di progettazione del costo del cambiamento e non focalizzarsi unicamente sul costo di produzione a regime. Una delle soluzioni adottate a tal proposito prevede l'utilizzo di stampi da prototipizzazione in quanto, nonostante siano caratterizzati da un tasso di usura più elevato, presentano un costo di molto inferiore ai normali stampi, permettendo un ammortamento dell'investimento in tempi più brevi e, di conseguenza, una maggior tasso di innovazione.

Architettura modulare Object-Oriented. In un'architettura modulare ogni blocco fisico implementa una o uno specifico insieme di funzioni e ha relazioni ben definite con gli altri blocchi. L'architettura modulare consente quindi di fare cambiamenti progettuali su un singolo blocco, senza dover modificare gli altri per avere il funzionamento corretto del prodotto, e di progettare i diversi blocchi in modo indipendente gli uni dagli altri. (Architettura Modulare, s.d.)

²⁶ Joe Justice è consulente presso Scrum, Inc., relatore TEDx e coach per team agili nel settore di produzione e sviluppo hardware. È il fondatore del Team Wikispeed, un'azienda di prototipazione automobilistica "verde" interamente basata su volontari Scrum, con l'obiettivo di cambiare il mondo in meglio.

²⁷ *Miles per gallon gasoline equivalent* è un'unità di misura che indica il percorso medio per unità di carburante consumato. 100 MPGe corrispondono a 2,35 l/100km.

Test Driven Development. Consiste in una tecnica che trae origine dal mondo dello sviluppo software, il cui obiettivo è definire in anticipo rispetto allo sviluppo stesso i parametri e le condizioni affinché ciò che andiamo a sviluppare sia considerato accettabile. L'approccio da seguire per progettare un nuovo componente è il seguente:

- Creare il test che si desidera venga superato. Se possibile lo si rende automatico in modo da eliminare i costi delle riparazioni nelle iterazioni successive.
- Creare il più semplice design che permette di superare il test.
- Iterare il design, migliorandolo incrementalmente, finché il risultato non è soddisfacente.

Contract-First Design. Al momento di progettare una soluzione per i diversi moduli che caratterizzano il prodotto occorre inizialmente definire le interfacce di collegamento agli altri moduli in funzione dei parametri conosciuti, come ad esempio fattori di carico o requisiti di resistenza; successivamente disegnare solamente le connessioni e non i componenti individuali e lasciare spazio a possibili evoluzioni, prevedendo interfacce in eccesso.

Iterazione del Design. Lo sviluppo di prodotti fisici ha bisogno di un punto di vista diverso sulle iterazioni rispetto al software, in quanto per iterare il design occorre:

- Creare un test che il componente deve superare.
- Disegnare il componente nella maniera più semplice che permetta di superare il test.
- Migliorare il design del componente in modo che risulti più semplice e più elegante.
- Ripetere il processo (Iterazione del design) fino a quando le migliori aggiungono ancora valore al prodotto.

Realizzare un incremento di prodotto potenzialmente consegnabile ad ogni iterazione non è sempre possibile nello sviluppo di prodotti tangibili e risulta quindi necessario iterare più volte finché il design risulta accettabile.

Design Pattern. Consistono in un modo semplice per rappresentare la soluzione ad un problema noto. I pattern sono stati introdotti nell'architettura da Christopher Alexander per facilitare la comprensione di soluzioni valide a problemi ricorrenti nell'ambito della costruzione di edifici. Lo sviluppo del software ha assimilato questa idea per diffondere le soluzioni migliori ai problemi più frequenti.

Integrazione continua. Tenere in considerazione svariate tipologie di soluzioni e, allo stesso tempo, realizzare un design coeso e consistente può risultare problematico. A livello ingegneristico, eXtreme Manufacturing

propone l'integrazione continua con esecuzione frequente di test automatici di tipo Test Driven Development, sfruttando ad esempio il *Metodo degli Elementi Finiti*²⁸. In questo modo ciascun membro del team può sviluppare soluzioni di miglioramento in parallelo e i test consentono di individuare l'alternativa migliore.

Rilascio continuo. Il principio del rilascio continuo richiede l'uso di linee di montaggio flessibili, in grado di produrre diverse tipologie di pezzi in pochi giorni. La flessibilità operativa richiesta può essere raggiunta adottando macchinari di prototipazione rapida di tipo sia additivo che sottrattivo, i quali dovranno prevedere dei sistemi di movimentazione che consentano di modificare il layout e il flusso operativo del processo in funzione degli obiettivi dell'iterazione in corso. Il principio del rilascio continuo prevede quindi di creare una prima versione del prodotto durante la prima iterazione e, durante gli Sprint successivi, di migliorare iterativamente il design grazie ai feedback di clienti, utenti e stakeholder. Si passerà quindi da delle prime versioni più grezze e artigianali ad un concept più raffinato e definito, fino a raggiungere rapidamente l'obiettivo (Sammicheli, 2018).

Scalare organizzazioni. Scrum, come originariamente descritto nella Scrum Guide, è un framework per sviluppare, consegnare e mantenere prodotti complessi da un singolo team. Fin dalla nascita, il suo uso si è esteso alla creazione di prodotti, processi, servizi e sistemi che richiedevano il lavoro di più team. Tale processo di estensione del framework da un singolo team all'intera organizzazione, viene descritto nella guida Scrum@Scale²⁹. Il metodo eXtreme Manufacturing sfrutta le stesse logiche di scaling.

Gestire fornitori. La gestione dei fornitori viene definita in (Sammicheli, 2018) come segue: "Ogni componente proveniente da fornitore esterno è collegato al prodotto con un adattatore che fornisce una interfaccia stabile, ovvero che non cambia anche nel caso in cui cambi il fornitore. È quindi possibile variare tra fornitori esterni, prototipi auto-costruiti e ordinativi di larga scala ad un costo minimo. Per rendere più rapidi gli acquisti, è possibile ordinare prodotti dall'esterno per caratteristiche anziché per specifiche tecniche, in questo modo anche il fornitore può essere creativo e creare innovazione."

²⁸ Tecnica numerica atta a cercare soluzioni approssimate di problemi descritti da equazioni differenziali alle derivate parziali riducendo queste ultime a un sistema di equazioni algebriche. In generale, il metodo agli elementi finiti si presta molto bene a risolvere equazioni alle derivate parziali quando il dominio ha forma complessa (come il telaio di un'automobile o il motore di un aereo), quando il dominio è variabile (per esempio una reazione a stato solido con condizioni al contorno variabili), quando l'accuratezza richiesta alla soluzione non è omogenea sul dominio (in un crash test su un autoveicolo, l'accuratezza richiesta è maggiore in prossimità della zona di impatto) e quando la soluzione cercata manca di regolarità.

²⁹ <https://www.scrumatscale.com>

2.3 Agile Stage-Gate

Il modello ibrido Agile Stage-Gate (o Scrum Stage-Gate) nasce dalla combinazione del modello Stage-Gate con metodologie Agile o il framework Scrum, con l'obiettivo di combinare i punti di forza di entrambi gli approcci.

Il modello Stage-Gate viene formalizzato da Robert G. Cooper negli anni '80 per incrementare il tasso di successo dei nuovi prodotti e ridurre il time to market dello sviluppo. Tale metodologia Plan-Driven di gestione dei progetti è stata, e lo è tutt'ora, ampiamente adottata da aziende manifatturiere nell'ambito dello sviluppo di prodotti.

(Cooper R. G., 1988) presenta la seguente definizione: “Lo Stage-Gate descrive un modello in cui il processo di sviluppo del prodotto, dalla generazione dell'idea al lancio sul mercato, è suddiviso in fasi discrete, ognuna con compiti definiti e risultati prescritti.”

I Gate che precedono ogni fase rappresentano dei momenti formali di decisione di investimento, in cui si definisce se continuare con il progetto o bloccarlo (Go/Kill).

I vantaggi del modello Stage-Gate rispetto al modello informale, in accordo con (Cooper, Edgett, & Kleinschmidt, 2002), includono:

- Maggiore velocità di sviluppo;
- Migliore qualità;
- Maggiore disciplina;
- Migliori prestazioni complessive;

(Zhang, 2012) afferma che l'obiettivo di molti degli strumenti associati al processo Stage-Gate è “ridurre le iterazioni prevedendo il processo dall'inizio”.

Come descritto nell'introduzione dell'elaborato, i sistemi di gestione dei progetti con metodologie che prevedono la definizione anticipata delle specifiche del prodotto non sono adatte in condizioni di elevata turbolenza ed incertezza ambientale.

Tale ipotesi è confermata in (Cooper & Sommers, 2018) come segue: “Il modello di Gating tradizionale non consente modifiche, in quanto la definizione del prodotto e il piano di sviluppo sono bloccati, creando problemi di gestione delle modifiche a valle. I sistemi di gate sono semplicemente troppo lineari e troppo rigidi per adattarsi efficacemente ai mercati instabili e in rapida evoluzione e alle esigenze dei clienti che guidano i nuovi prodotti di oggi.”

In accordo con (Biazzo & Filippini, 2018), tali condizioni possono essere generate da molteplici fattori:

- Le richieste del cliente sono in rapida evoluzione e i problemi da risolvere non sono ancora ben precisi;
- Le tecnologie di base del prodotto possono trovarsi in uno stadio evolutivo ad alta turbolenza;
- La concorrenza è intensa;
- Le soluzioni architetturali presentano un ampio spazio di possibilità e sperimentazione.

Le decisioni che vengono ritenute definitive possono causare delle rilavorazioni indesiderate nelle fasi avanzate del progetto, causando delle perdite di performance in termini di costi e tempi del progetto.

Un ulteriore aspetto riguarda il fenomeno della devianza burocratica, ovvero l'eccessiva richiesta di deliverable e reportistica al fine del controllo, i quali costituiscono dei punti di discontinuità, risultando in contrasto con lo scorrere del flusso delle attività a valore.

A partire da queste lacune, alcune aziende hanno iniziato ad integrare al modello tradizionale pratiche di gestione dei progetti provenienti dall'industria software, come Scrum, con l'obiettivo di rendere il processo di sviluppo più flessibile. Questo ha portato alla nascita di modelli ibridi.

2.3.1 La gestione dei progetti tramite Hybrid Stage-Gate

Nella gestione dei progetti di sviluppo, e più in generale nella pratica dell'Operations Management, la pianificazione e il controllo sono suddivisi in tre livelli: strategico, tattico ed esecutivo (Stevenson, 2005).

Le decisioni di lungo periodo, che riguardano l'intera durata del progetto, compongono il livello strategico. All'interno di questo livello sono identificate le milestones (momenti di integrazione e di condivisione critica con tutto il team) del progetto, che consentono al top management di monitorare l'avanzamento del progetto e decidere di continuare in base ai risultati ottenuti o previsti.

A livello tattico viene definita la pianificazione delle attività e delle risorse con orizzonte temporale settimanale/mensile, la quale deve integrarsi con le decisioni prese a livello strategico. Il livello esecutivo riguarda le decisioni sulle attività giornaliere da completare.

La metodologia Hybrid Stage - Gate [Figura 8] permette di conciliare il bisogno di una pianificazione su più livelli prevedendo la gestione del livello tattico e del livello esecutivo tramite l'utilizzo del framework Scrum, oppure altre tipologie di approcci Agile. In particolare, la pianificazione a livello tattico è supportata dagli eventi di Sprint Planning e dallo Sprint Review, mentre la pianificazione a livello esecutivo dal Daily Scrum³⁰.

Suddividere i vari stage del processo di sviluppo in Sprint significa frammentare periodi di decine di mesi in finestre temporali con una durata di massimo quattro settimane. Così facendo è possibile migliorare la pianificazione e suddivisione del lavoro e soprattutto diminuire la variabilità del processo (Cooper & Sommers, 2018).

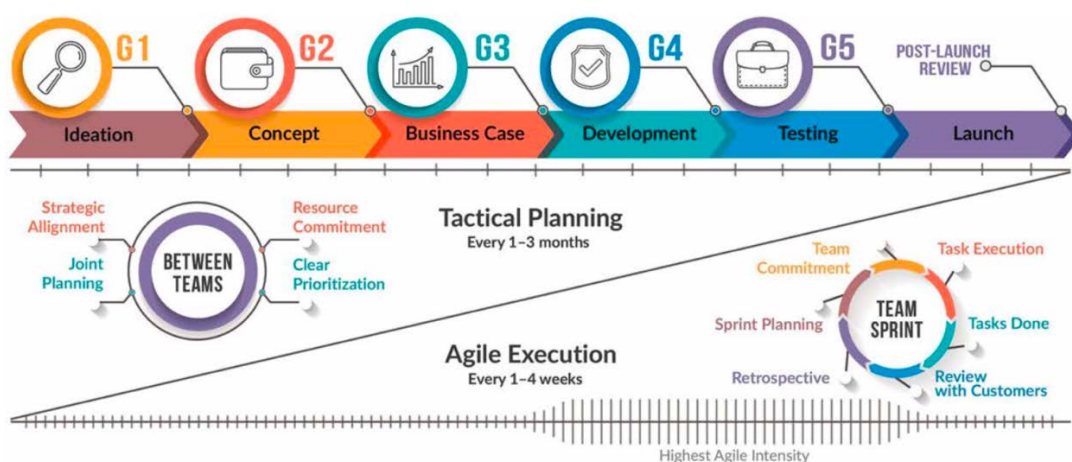


Figura 8 - Hybrid Agile Stage Gate (Cooper & Sommers, 2018)

³⁰ Entrambi gli eventi di Scrum sono descritti all'interno del primo capitolo

2.3.2 Benefici e sfide dei modelli ibridi

I benefici della combinazione del modello Stage-Gate con Scrum, o metodologie Agile, basandosi su casi studio applicativi del modello ibrido sono i seguenti:

- Maggior Flessibilità;
- Permette di affrontare l'incertezza;
- Permette di realizzare il prodotto giusto: numerosi studi rivelano che molti nuovi prodotti falliscono o non offrono il valore promesso quando vengono introdotti sul mercato ed il motivo principale del fallimento è rappresentato da una mancanza di comprensione delle esigenze del cliente. Per incrementare la probabilità di successo risulta necessario l'adozione di Sprint iterativi che prevedano un maggior coinvolgimento del mercato, in modo da essere certi di realizzare ciò che viene richiesto (Vedsmand, 2016);
- Accelera lo sviluppo: abilita ad una riduzione del time to market (che rappresenta un obiettivo condiviso nel contesto odierno) tramite l'approccio altamente iterativo, time-boxed e rapido, focalizzato su come sviluppare incrementi di prodotti funzionanti (Vedsmand, 2016).
- Migliora della produttività, della comunicazione e del coordinamento tra i membri del team di progetto;
- Maggiore attenzione ai progetti, con conseguente migliore definizione delle priorità di tempo e sforzi;
- Incrementa il morale del Team.

In accordo con (Cooper & Sommers, 2018), diversi casi studio hanno confermato anche la presenza di problemi e sfide durante l'implementazione dell'Agile Stage-Gate nello sviluppo di prodotti tangibili.

Risolvere le incoerenze. I potenziali conflitti all'interno del modello ibrido diventano evidenti nella lotta per bilanciare la fluidità a breve termine con la pianificazione a lungo termine. Vi è inoltre la necessità del modello Stage-Gate di una definizione di prodotto concreta contro la definizione di prodotto in costante evoluzione di Scrum (Cooper & Sommers, 2018).

Scrum e le metodologie Agili non prediligono piani di lungo termine, in quanto il cambiamento provocherebbe una modifica a tali piani, che quindi non avrebbe senso definire. Il focus sulla pianificazione a breve termine e in costante evoluzione provoca disorientamento nelle aziende manifatturiere, le quali si trovano in difficoltà nella stima dei costi e dei tempi dello sviluppo, necessari a garantire l'approvazione del progetto. Tale difficoltà può essere superata definendo un programma di attività a livello macro per ciascuna fase basato sulla conoscenza attuale del progetto e allo stesso tempo sensibilizzando il management riguardo al cambiamento. Le stime dei costi e delle tempistiche dei progetti hanno sempre subito un cambiamento nella

maggior parte dei progetti di sviluppo che adottavano metodologie Plan-Drive. Si tratta di modifiche indesiderate intrinseche del modello di gestione del progetto. Anche nell'Agile Stage-Gate si andrà incontro a delle modifiche delle stime di costi e di tempi, ma in questo caso il cambiamento è riconosciuto e abbracciato in anticipo.

Come ha osservato il direttore senior della ricerca e sviluppo di Danfoss³¹, i manager devono ricordare che: “Agile Stage-Gate consente convalide tempestive e frequenti dei clienti per la progettazione di prodotti fisici e virtuali. Il cambiamento principale è che le specifiche di progettazione non sono più fissate in anticipo, ma vengono continuamente adattate attraverso le iterazioni di progettazione: non esiste più un blocco del progetto pre-sviluppo.” (Cooper & Sommers, 2018).

La gestione del processo di modifica delle caratteristiche del prodotto avviene attraverso il Product Backlog, uno degli artefatti di Scrum.

Come nella pianificazione e nel budgetting, le realtà manifatturiere devono imparare a gestire più ambiguità di quella con cui potrebbero essere a proprio agio nella definizione del prodotto, andando avanti sulla base di una definizione di prodotto di alto livello, anche se in qualche modo un po' provvisoria (Cooper & Sommers, 2018).

Trovare le risorse. La sfida principale per molte aziende è trovare le risorse necessarie per supportare un approccio Agile, in particolare i team di progetto dedicati richiesti da una piena implementazione del sistema. Gli eventi di Scrum sono molto più difficili da portare a termine in modo efficace se i membri del team non interagiscono tra loro con una comunicazione regolare o sono distribuiti su più fronti. L'impossibilità di dedicare le risorse completamente ad un singolo progetto, per mancanza di skill o di personale, ha obbligato alcune aziende a raggiungere un compromesso, come ad esempio limitare il tempo dedicato ad altri progetti al 20-30% del tempo totale di lavoro.

Definizione di “fatto” di uno Sprint. Come già descritto all'inizio del secondo capitolo, non sempre il risultato di uno Sprint può consistere in un incremento potenzialmente rilasciabile nel caso di prodotti tangibili.

Nello sviluppo software questo problema non sussiste, o almeno raramente, in quanto per sua natura è infinitamente divisibile e quindi ad ogni singolo Sprint può essere realizzata e rilasciata una parte di codice. Un prodotto tangibile tecnologicamente complesso, al contrario, è difficilmente suddividibile in modo tale che durante lo sviluppo ci siano degli incrementi di funzionalità.

³¹ Danfoss è una multinazionale danese nata nel 1933 a Nordborg, nello Jutland meridionale. Produce prodotti refrigeranti per alimenti, aria condizionata, riscaldamento di edifici, controllo di motori elettrici, compressori di gas, e azionamenti a frequenza variabile.

Le realtà manifatturiere hanno superato l'ostacolo attribuendo una nuova definizione di "fatto". Il risultato di uno Sprint "fatto" in questo contesto non è necessariamente un prodotto funzionante; è semplicemente un risultato tangibile del lavoro completato in quello Sprint. Ad esempio, nelle fasi precedenti del front-end, la definizione di "fatto" può essere un caso aziendale o i risultati di esperimenti preliminari o uno studio sulla voce del cliente. A partire dalla fase di sviluppo, tuttavia, uno Sprint "Fatto" di solito produce qualcosa di fisico a cui il cliente può rispondere e la direzione può vedere: disegni di progettazione, un'animazione al computer, un prodotto virtuale, un modello grezzo, in breve una versione del prodotto.

Anche ottenere un rapido feedback dei clienti alla fine dello Sprint può essere una sfida. Per il software, ottenere feedback comporta l'invio della nuova funzionalità ai clienti, in genere elettronicamente, e la richiesta di feedback da parte dello stesso canale. Ma i prodotti hardware non possono essere inviati via e-mail o scaricati. E alcuni clienti non vogliono spostarsi così velocemente o non possono eseguire test così rapidamente. In LEGO e Danfoss, interi Sprint sono dedicati alla ricerca della convalida del cliente. Altre aziende chiedono ai clienti strategici di impegnarsi a prendere parte al processo di sviluppo prima dell'avvio del progetto. Il vantaggio per il cliente nell'investire il proprio tempo e gli sforzi è che il prodotto finale è adattato esattamente alle sue esigenze e immediatamente implementabile nei propri sistemi (Cooper & Sommers, 2018).

Dove implementare Scrum. Una delle questioni chiave dell'Agile Stage-Gate riguarda la definizione delle fasi del processo di sviluppo nelle quali adottare il framework e per quali categorie di progetto.

In letteratura viene sottolineato che l'applicazione del modello ibrido risulta più complementare alle fasi "tecniche" dei progetti. Da un lato perché all'interno di queste fasi è più probabile che il team sia in grado di generare con una maggior attenzione degli incrementi di prodotto funzionanti o dimostrabili e dall'altro perché le persone tecniche è più probabile possano focalizzarsi su un singolo progetto, facilitando la pianificazione delle risorse.

L'applicazione del modello ibrido può però essere allargata anche alle fasi iniziali dei progetti, corrispondenti alle fasi di ideazione e di concept, comportando però una maggior difficoltà nel garantire un incremento di prodotto dimostrabile al termine di ciascuno Sprint.

2.3.3 Caratteristiche dei processi di sviluppo con modelli ibridi

In accordo con (Cooper R. G., 2014), l'adozione di un modello ibrido di gestione dei processi di sviluppo consente di creare un sistema denominato "3A", in quanto presenta le seguenti caratteristiche:

Adattabile e Flessibile. Molto spesso il concept di prodotto non può essere completamente congelato nelle fasi iniziali del progetto. Per tale motivo un processo di sviluppo che adotta un modello ibrido è *Adattabile* in quanto gestisce un concept aperto, per cui è definito meno del 50% del prodotto, e in modo iterativo converge verso il concept definitivo. Questo approccio è stato una delle chiavi del successo nello sviluppo del prodotto per Steve Jobs in Apple, il quale affermava che *"Non puoi solo chiedere ai clienti che cosa vogliono e poi provare a darglielo. Per quando l'avrai costruito, vorranno qualcosa di nuovo."*

La natura del nuovo modello ibrido deve essere inoltre *Flessibile*, ciò significa che non vengono definite attività o deliverable prestabiliti per ogni progetto, ma in base al prodotto che si sta sviluppando e al contesto del mercato, verranno definite in modo specifico per quel singolo progetto.

Agile. L'adozione del framework Scrum al livello esecutivo e tattico della pianificazione del progetto di sviluppo consente di muoversi da milestone a milestone all'interno dello stage in modo agile, attraverso continui cicli di iterazione e apprendimento, riducendo la variabilità e gli sprechi presenti tipicamente all'interno del processo tradizionale.

Accelerato. Il nuovo modello ibrido è focalizzato a ridurre al minimo il Time to Market dei progetti di sviluppo. Questo è garantito da alcune caratteristiche come la presenza di un team inter-funzionale dedicato al progetto. Altre caratteristiche a supporto dell'accelerazione sono la sovrapposizione e la parallelizzazione dei vari task all'interno degli stage. Spesso gli stessi stage sono leggermente sovrapposti tra loro, rendendo il concetto di stage meno rigido rispetto ai modelli tradizionali.

2.3.4 Criteri flessibili ai Gate

Il modello tradizionale di Stage-Gate formalizzato da Cooper negli anni '80-'90 è caratterizzato dalla definizione di criteri di carattere economico finanziario, come ROI e flusso di cassa, per sancire la continuazione o la sospensione del progetto di sviluppo in analisi.

Essendo i progetti di sviluppo caratterizzati da un'incertezza intrinseca dovuta alla loro natura innovativa, l'utilizzo di indicatori economici come metro di giudizio non garantisce un'analisi corretta e accurata delle potenzialità del progetto, in quanto la maggior parte delle componenti di tali indicatori rappresentano solamente delle stime previsionali o comparative.

In progetti fortemente innovativi o in contesti turbolenti risulta infatti difficile fare previsioni accurate su quali possano essere le previsioni di vendita, l'andamento del mercato o le risposte dei competitor per definire i ricavi e i flussi di cassa futuri.

Risulta quindi necessario fondare la definizione dei criteri di Go/Kill su regole più flessibili, che comprendano anche criteri economico-finanziari ma non solo, integrando l'adattamento strategico del progetto, i potenziali benefici futuri in termini di reputazione o alleanze, il vantaggio competitivo e il potenziale di mercato.

Capitolo 3.

Il Caso studio: FILA Industria Chimica s.p.a.

*“Non sempre cambiare equivale a migliorare,
ma per migliorare bisogna cambiare.”*

Winston Churchill

Nel seguente capitolo viene esposto brevemente il processo che ha portato alla definizione di un modello Hybrid Stage - Gate per la gestione del processo di development in FILA Industria Chimica s.p.a., nella quale ho svolto il periodo di stage per conto della società di consulenza auxiell s.p.a.

In seguito, vengono descritti l'organizzazione del nuovo processo e gli standard adottati, ed è inoltre riportato un esempio reale di svolgimento di uno Sprint.

Prima di iniziare l'esposizione degli esempi e degli standard adottati per l'applicazione di Scrum nello sviluppo di prodotti tangibili, procederò con una breve descrizione delle società che mi hanno dato l'opportunità di svolgere l'esperienza formativa.

3.1 auxiell s.p.a

«Leading by Example»

auxiell³² è una società di consulenza *lean* con sede a Padova nata nel 2005 dall'esperienza maturata nel supportare la trasformazione dell'intera supply chain di Unox s.p.a.³³

Per supportare le organizzazioni nel raggiungere l'eccellenza e diventare un esempio, auxiell sviluppa il framework dell'*auxiell lean system*: un sistema di principi, tecniche, metodi e strumenti che permette il miglioramento continuo dei processi di generazione di Valore per il Cliente.

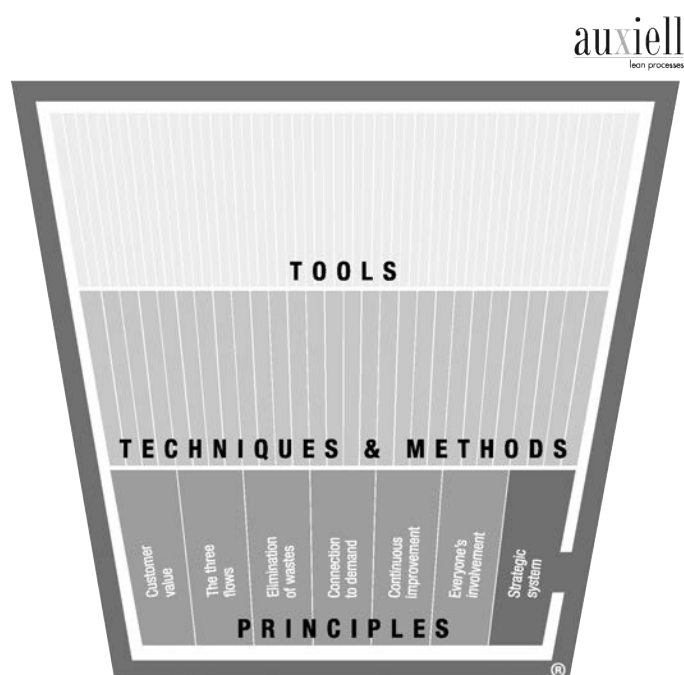


Figura 9 - auxiell lean system

La particolare struttura del framework [Figura 9] consente di focalizzare appieno l'attenzione sulla differenza tra “fare lean”, ovvero applicare degli strumenti e tecniche sfiorando solamente la superficie del problema, ed “essere lean”, ovvero comprendere ed applicare i principi del lean system per guidare profondamente la trasformazione dei processi.

³² D'ora in poi all'interno dell'elaborato verrà usato solamente auxiell al posto di auxiell s.p.a.

³³ Unox Spa, azienda produttrice di forni professionali con sede a Cadoneghe (PD), è stata una delle prime PMI ad eseguire una trasformazione completa e sistemica dei processi interni e dell'intera *supply chain*.

Il *core business* di auxiell è fondato sulla *complete lean transformation*, ovvero la trasformazione sistemica dei principali processi di generazione di valore, le cosiddette 4D³⁴, perseguita adottando un approccio scientifico fondato sul ciclo SPDCA³⁵ [Figura 10].

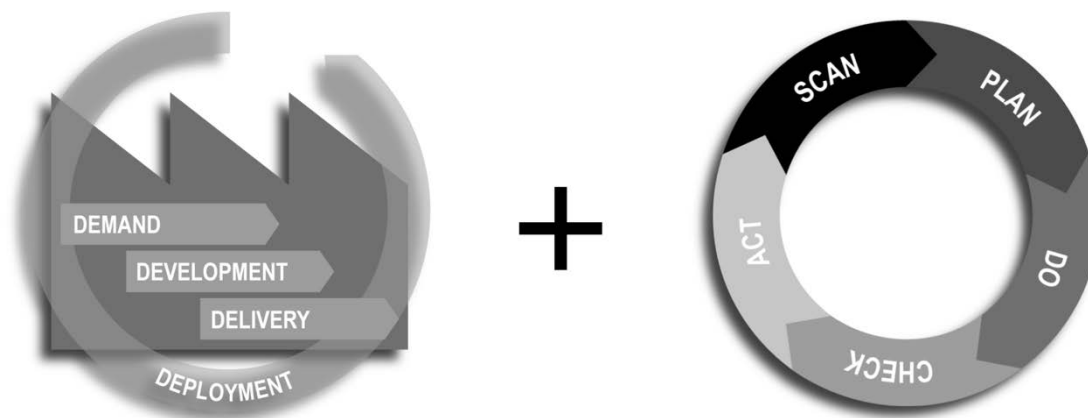


Figura 10 - 4D e ciclo SPDCA - <https://www.auxiell.com>

Nell'immagine sottostante [Figura 11] sono illustrate le fasi che caratterizzano i progetti di *complete lean transformation* delle aziende descritte nei casi aziendali. Ciascuna di queste fasi rappresenta un prodotto della *Value Proposition* di auxiell.

	SCAN	PLAN	DO	CHECK	ACT
DELIVERY	LEAN GAP ANALYSIS	LEAN ROADMAP	DELIVERY OVERALL TRANSFORMATION	LEAN CHECK UP	
DEVELOPMENT	PROCESS OVERVIEW	LEAN ROADMAP	LEAN INDUCTION	LEAN CHECK UP	LEAN WORKOUT
DEMAND	PROCESS OVERVIEW	LEAN ROADMAP	LEAN STUDY TOUR	LEAN CHECK UP	LEAN ACCOUNTING
DEPLOYMENT	LEAN GAP ANALYSIS	LEAN ROADMAP	HOSHIN KANRI	LEAN CHECK UP	

Figura 11 - Value Proposition di auxiell - <https://www.auxiell.com>

³⁴ Rappresentano i 4 macro-processi di generazione del valore: Delivery (processo logistico-produttivo: trasforma ordini dei clienti e risorse da trasformare in prodotti finiti), Development (processo di sviluppo prodotto-processo: trasforma idee in processi profittevoli e conoscenza usufruibile), Demand (processo di vendita: trasforma potenziali clienti in clienti), e Deployment (processo di sviluppo strategico: trasforma la strategia aziendale in un set di programmi per renderla operativa).

³⁵ Il ciclo SPDCA è un'evoluzione del ciclo PDCA, nel quale si va ad individuare una fase iniziale denominata Scan. In particolare, le fasi operative del ciclo SPDCA sono: *Scan* - analisi della situazione di partenza; *Plan* - individuazione delle opportunità di miglioramento, progettazione dello stato futuro e definizione dell'action plan per raggiungere tale condizione; *Do* - implementazione; *Check* - controllo dei risultati; *Act* - stabilizzazione ed estensione ad altri processi.

3.2 Introduzione a FILA Industria Chimica s.p.a.

«Surface Care Excellence, Worldwide»

“Rappresentare l’eccellenza nel settore industriale di riferimento e supportare i nostri clienti fornendo in tutto il mondo soluzioni innovative, sicure ed eco-compatibili per la cura e il trattamento di ogni tipo di superficie”



Figura 12 - Logo FILA Industria Chimica s.p.a. - www.filasolutions.com

FILA Industria Chimica s.p.a.³⁶ nasce nel 1943 a San Martino di Lupari (PD) come produttore e fornitore di lucido per calzature e detergenti domestici, diventando in breve tempo riferimento in Italia per la produzione di lucido da scarpe di alta qualità e venendo scelta anche dall’esercito italiano.

Negli anni '70, cogliendo le opportunità di un mercato in evoluzione e recependone i bisogni non ancora soddisfatti, l’azienda evolve il suo *core business* nella produzione di prodotti professionali per la cura delle superfici.

Inizialmente il materiale di riferimento è il cotto, ma grazie agli investimenti in ricerca e sviluppo, l’ampiezza delle soluzioni offerte dall’azienda si allarga a tutti i materiali per pavimento e rivestimento.

Oggi FILA ha raggiunto l’eccellenza mondiale nei sistemi per la protezione e manutenzione di tutte le superfici.

La sede centrale, il centro produttivo e il centro ricerche sono tutt’ora localizzate a Padova, ma l’azienda rappresenta una realtà multinazionale con 6 filiali commerciali in Germania, Francia, Spagna, Gran Bretagna, USA ed Emirati Arabi e prodotti che vengono venduti in oltre 60 paesi al mondo.

³⁶ D’ora in poi all’interno dell’elaborato verrà usato solamente FILA.

3.2.1 L'offerta di FILA

La Value Proposition di FILA si articola in tre diverse Business Unit che operano contemporaneamente e in modo sinergico all'interno di tutti i mercati internazionali:

- Trade Division, a supporto della rete distributiva come gli applicatori professionisti, le ferramenta e i negozi specializzati, i colorifici e la grande distribuzione specializzata;
- Project Division, che si interfaccia con i professionisti, i prescrittori e i referenti tecnici dell'industria internazionale delle costruzioni, fornendo prodotti e soluzioni e un servizio di assistenza tecnica nella progettazione e in cantiere;
- Tech Division, che collabora con i produttori ed è specializzata nello studio di soluzioni per l'applicazione in linea di prodotti per la protezione dei materiali in ambito industriale.

3.2.2 La ricerca e sviluppo

Il carattere innovativo di FILA è ampiamente dimostrato dagli ingenti investimenti nella ricerca e sviluppo, che rappresentano il 5% del fatturato, e dalla realizzazione nel 2018 del nuovo centro all'avanguardia di ricerca e sviluppo [Figura 13]. Tra le principali attività che vengono svolte all'interno del centro di ricerca ci sono:

- Analisi e studio dei materiali;
- Selezione delle migliori materie prime e dei fornitori;
- Sperimentazione quotidiana di nuove soluzioni e prodotti;
- Test e collaudo dei nuovi prodotti;
- Controllo qualità dei prodotti realizzati, certificata dai maggiori enti internazionali.



Figura 13 - Centro di ricerca e sviluppo di FILA

3.2.3 La necessità di evolvere

Nel 2018 il management familiare dell'azienda partecipa al *Lean Japan Study Tour* guidato da CUOA³⁷ con la partecipazione di auxiell e acquisisce la consapevolezza della necessità di una nuova evoluzione per rimanere al passo con le sempre più esigenti ed eterogenee richieste del mercato.

A inizio 2019 avvia quindi un percorso di *complete lean transformation*, inizialmente nel processo di *delivery*, con un programma strategico di trasformazione di tutti i processi aziendali.

Successivamente, verso la fine del 2019, l'azienda decide di iniziare la trasformazione dei processi di *development* e *deployment*.

³⁷ La Fondazione CUOA è una scuola di management, che svolge attività di formazione e sviluppo della cultura imprenditoriale e manageriale. Ha sede nella Villa Valmarana Morosini ad Altavilla Vicentina in provincia di Vicenza.

3.2.3 Lean Gap Analysis® e Lean RoadMap®

La Lean Gap Analysis® rappresenta la prima macro-fase del progetto di *complete lean transformation* e coincide con la fase di Scan del ciclo SPDCA.

Essa consiste nel creare una fotografia della situazione di partenza dell'organizzazione, ovvero dello stato "as is", con l'obiettivo di definire chiaramente i gap tra le performance attuali dei processi di generazione di Valore del cliente e quelle dei processi ottimizzati sviluppati in accordo con i principi del Lean System.

La fase successiva del progetto coincide con la fase di Plan del ciclo SPDCA e prende il nome di Lean RoadMap®. L'output di questa seconda fase è un piano d'azione in cui si evidenzia la sequenza di implementazione delle più adeguate tecniche che consentano di raggiungere in modo efficiente ed efficace lo stato futuro del processo in analisi.

Al fine di comprendere le motivazioni che hanno spinto ad implementare all'interno dell'organizzazione un framework di pianificazione, ritengo sia opportuno descrivere brevemente il percorso di trasformazione a partire dallo stato "as is" di FILA.

3.2.3.1 Value Stream Mapping

Prima di iniziare ad agire nei processi di un'organizzazione, risulta fondamentale mapparli in modo da comprenderli ed essere in grado di identificare con chiarezza gli sprechi all'interno dei processi stessi. Ciò risulta necessario in quanto le risorse coinvolte spesso hanno difficoltà a riconoscerli o a vederli tutti, poiché ormai abituate a convivervi.

La *Value Stream Mapping* è una tecnica che consente di vedere e descrivere attraverso un linguaggio standardizzato il processo e le sue logiche.

Tale tecnica permette quindi di avere una visione globale dell'intero processo e viene utilizzata nella loro riprogettazione. Le fasi principali che la caratterizzano sono le seguenti:

- Definizione del Current State del processo (vedi paragrafo 3.2.3.3): rappresentazione grafica dello stato attuale del processo con evidenza dei *muda*³⁸;
- Definizione del Future State del processo (vedi paragrafo 3.2.3.5): rappresentazione grafica dello stato futuro del processo, definito a partire dalla Current State e dalle linee guida di un processo che segue i principi del Lean System;

³⁸ Muda (無駄) è un termine giapponese che identifica attività inutili o che non aggiungono valore o improduttive.

3.2.3.2 Definizione del progetto pilota

Prima di procedere con la mappatura dello stato “as is” del processo di Development di FILA, è stata organizzata una riunione con il team con l’obiettivo di definire il processo pilota oggetto di analisi.

Le tipologie di processo di sviluppo all’interno di FILA sono le seguenti:

- Sviluppo nuovo prodotto;
- Modifica prodotto;
- Personalizzazione;
- Ricerca.

Partendo dal bisogno di trasformare il macro-processo di development, il team ha scelto come processo pilota lo *sviluppo nuovo prodotto*.

I driver che hanno portato alla scelta del processo di sviluppo nuovo prodotto come progetto pilota sono stati:

- Il bisogno di standardizzare e trasformare tutte le fasi del processo, a partire dalla definizione della specifica di prodotto fino al lancio in produzione;
- La scelta di un progetto che potesse avere un impatto rilevante nelle performance aziendali;
- La scelta di un progetto prioritario per le persone direttamente coinvolte, che permettesse quindi di aumentare l’engagement delle stesse.

Gli effetti dello svolgimento del progetto pilota nel personale direttamente coinvolto nel processo risultano fondamentali. Attraverso il progetto essi hanno la possibilità di entrare in contatto e prendere confidenza con i principi e le tecniche del *lean system*, di sviluppare un occhio critico circa i problemi e le possibili soluzioni e di sentirsi parte integrante e trainante del cambiamento, in modo da poterlo portare avanti anche in seguito al progetto di trasformazione. Quest’ultimo punto risulta determinante, in quanto le soluzioni che vengono implementate devono essere sviluppate assieme a coloro che saranno coinvolti in prima persona nel loro utilizzo, in accordo con il principio del *coinvolgimento di tutti*³⁹, pena il possibile fallimento del progetto.

Il progetto pilota è inoltre necessario per accrescere il know-how interno, in modo tale che al suo termine sia possibile un’estensione rapida ed efficace agli altri processi aziendali.

³⁹ Il principio del *coinvolgimento di tutti* è il sesto principio dell’*auxiliell lean system*. Tale principio va a sottolineare che per costruire con successo un sistema lean e una *lean organization* risulta necessario creare un cambiamento culturale in tutte le persone dell’organizzazione.

3.2.3.3 Current State Map

Di seguito [Figura 14] è presentata la mappatura dello stato corrente (precedente al progetto di trasformazione) del processo di sviluppo nuovo prodotto di FILA.

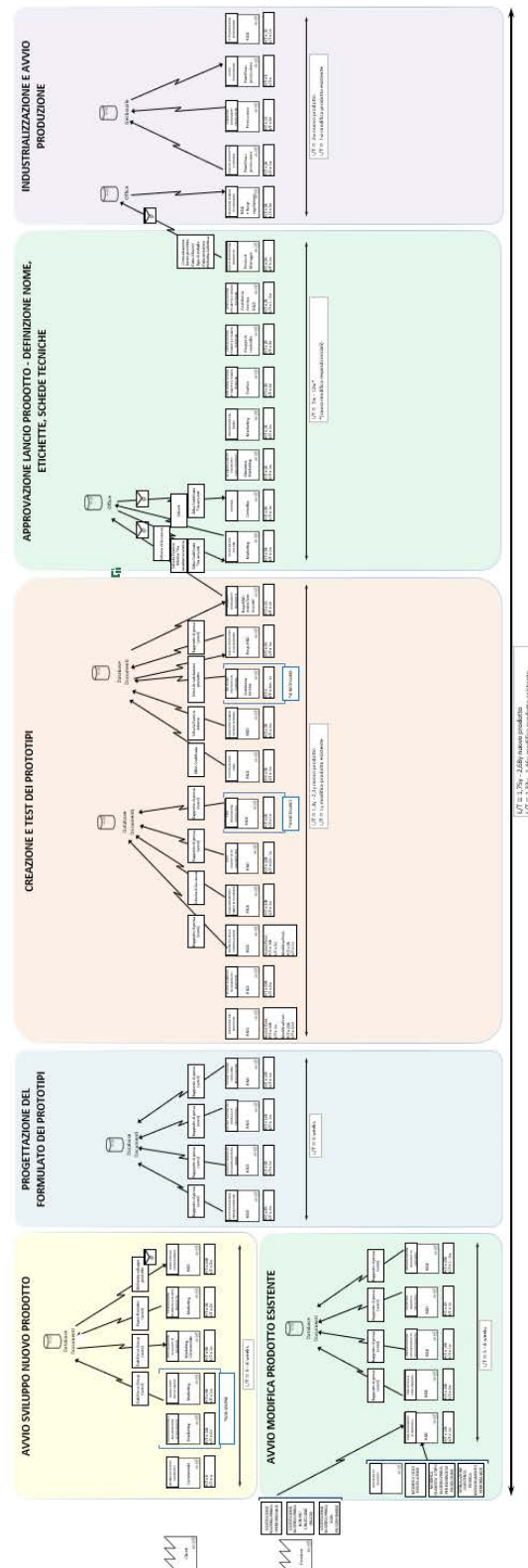


Figura 14 - Current State Map e Muda Identification del processo di sviluppo di FILA - Fonte: auxiell

3.2.3.4 Muda Analysis

La fase successiva alla mappatura dello stato “as is” del processo è l’analisi degli sprechi identificati tramite la tecnica della *Muda Analysis*. Tale tecnica prevede la suddivisione delle attività non a valore identificate nella mappatura del processo in cinque categorie, denominate 4M+E, rappresentative delle parti che compongono un processo. Esse sono:

- Men, manodopera;
- Materials, materie prime e ausiliarie;
- Methods, procedure o prassi operative;
- Machines, strumenti di lavoro o di misura utilizzati;
- Environment, condizioni ambientali esterne.

Dopo aver assegnato ciascun muda identificato alla componente delle 4M+E su cui ha impatto, si procede con la loro categorizzazione in funzione delle seguenti otto categorie di sprechi:

- Sovrapproduzione: produrre di più di ciò che serve alla fase successiva del processo, o prima rispetto a quando ce n’è effettivamente bisogno;
- Attese: inattività delle risorse da trasformare o trasformanti;
- Trasporti: movimentazione delle risorse da trasformare;
- Scorte: accumuli di risorse da trasformare, in trasformazione o trasformate;
- Movimenti: presenza nel flusso di lavoro di qualsiasi movimento non necessario;
- Processi non corretti: presenza di fasi di processo in eccesso dovute a inefficienze nelle risorse trasformanti o nella progettazione di prodotto o di processo;
- Difetti: generazione di output non aderenti alle specifiche tecniche o alle richieste del cliente;
- Conoscenza scartata: mancata formalizzazione e condivisione della conoscenza aziendale informale.

È possibile contestualizzare le otto categorie di sprechi in relazione alla specificità del processo di development come segue:

- Sovrapproduzione: eseguire un’attività superflua o cominciare un’attività in anticipo;
- Attese: attendere persone mancanti alle riunioni, attendere per approvazioni o attendere per l’output delle fasi precedenti;
- Trasporti: trasferimento di informazioni e responsabilità tra diverse funzioni;
- Scorte: lotti di output consegnati o da processare, materiale in attesa non ancora necessario;
- Movimenti: ricerca di persone o riunioni non a valore;

- Processi non corretti: mancata standardizzazione, multitasking e interruzioni, progetti in parallelo non completati;
- Difetti: reclami del cliente o disallineamento sviluppo-produzione;
- Conoscenza scartata: mancanza di “backup” della conoscenza delle risorse aziendali e mancata formalizzazione della conoscenza acquisita nel corso del progetto.

Dopo aver suddiviso i muda nelle diverse categorie, la Muda Analysis prevede l'applicazione della tecnica dei *5 Whys*. La tecnica dei “5 perché” è uno strumento sviluppato da Sakichi Toyoda e poi utilizzata in Toyota Motor Company Corporation nel TPS⁴⁰ che consente di identificare tramite un approccio scientifico la natura del problema alla base della generazione dei muda analizzati e, di conseguenza, ipotizzare una possibile soluzione.

“5 Whys is the basic of Toyota’s scientific approach [...] by repeating why five times, the nature of the problem as well as its solution become clear.”
Taiichi Ohno⁴¹

Di seguito [Figura 15] sono illustrati alcuni dei principali muda che hanno portato al bisogno di implementare un framework di pianificazione nel processo di development:

⁴⁰ Studiando i concetti fondanti del Fordismo e adattandoli al contesto di riferimento, tra il 1950 e il 1975 Toyota ha creato il modello produttivo che prende il nome di Toyota Production System: un modello per trattenere risorse attraverso l'eliminazione degli sprechi.

⁴¹ Taiichi Ōhno (Dalian, 29 febbraio 1912 – Toyota, 28 maggio 1990) è stato un ingegnere giapponese specializzato in meccanica, considerato il padre del Toyota Production System.

Muda	Category	1 Why?	2 Why?	3 Why?	4 Why?	5 Why?	Root cause	Possible solution
I progetti di sviluppo impiegano più tempo di quanto pianificato	Processi non corretti	Perché avrete l'esito del test sul prodotto va male e bisogna riprendere lo sviluppo						
I progetti di sviluppo impiegano più tempo di quanto pianificato	Processi non corretti	Perché gli input del progetto non sono chiari	Perché non è ben definito chi fa cosa				Perché non è ben definito chi fa cosa	
I progetti di sviluppo impiegano più tempo di quanto pianificato	Processi non corretti	Perché ci sono interruzioni					Perché ci sono interruzioni	
I progetti di sviluppo impiegano più tempo di quanto pianificato	Processi non corretti	Perché non si stabiliscono in maniera corretta le tempistiche	Perché non esiste una pianificazione che consideri il carico di lavoro, i tempi tecnici	Perché non c'è una valutazione "strategica", dall'alto, delle attività di sviluppo, si lavora principalmente in modalità "pompiaggio"			Perché non c'è una valutazione "strategica", dall'alto, delle attività di sviluppo, si lavora principalmente in modalità "pompiaggio"	Implementazione del sistema di pianificazione
Rilavorazioni degli output del processo di sviluppo, in base a modifiche dei requisiti fatte all'ultimo momento	Processi non corretti	Perché la direzione raccoglie informazioni dall'esterno in modo non strutturato, non condiviso e decide di modificare i requisiti iniziali	Perché la direzione decide di parzia, anche se ha dati interni a disposizione per decidere	Perché manca una strategia/piano di sviluppo prodotti biennale/triennale (roadmap) condiviso e definitivo	Perché manca un metodo per stabilire la strategia/piano di sviluppo basato su dati oggettivi		Perché manca un metodo per stabilire la strategia/piano di sviluppo basato su dati oggettivi	
Pianificazione delle risorse non fatta regolarmente	Processi non corretti							
Processi non corretti: assenza di visibilità sullo stato di avanzamento e sulle date di necessità delle varie fasi del processo	Processi non corretti							

Figura 15 - Muda Analysis Current State Map FILA - Fonte: auxiell

3.2.3.5 Future State Map

A partire dalla mappatura dello stato corrente, dall'analisi degli sprechi presenti e dalle linee guida di un processo che segue i principi del Lean System, viene definita la Future State Map del processo, della quale in seguito sono riportate la mappatura [Figura 16] e una breve descrizione di ciascuna delle fasi che la caratterizza.

Per la pianificazione delle attività a livello strategico del progetto di sviluppo nuovo prodotto, scelto come processo pilota dal team, si è adottato un modello Stage and Gate suddiviso in cinque stage con l'aggiunta di un processo ausiliario che si svolge in parallelo al processo principale.

Ricerca. La fase di ricerca è un processo il cui scopo è quello di testare la stabilità delle possibili miscele alla base dei prodotti, coinvolgendo i fornitori di nuove materie prime già prima che queste siano immesse sul mercato, per avere dei dati sperimentali in anticipo sullo sviluppo di nuovi prodotti.

La definizione di questo processo nasce dal bisogno di generare conoscenza usufruibile al team in fase di sviluppo di nuovi prodotti ed eliminare una delle attività time-consuming del vecchio processo di sviluppo, ovvero l'attesa dell'esito dei test di stabilità sulle materie chimiche e i miscelati, la quale può richiedere anche un anno.

Sviluppo Product Roadmap e Specifica di Prodotto. La prima fase del processo di sviluppo prevede le seguenti macro-attività:

- Il raccoglimento in input dei bisogni del cliente, da parte del team commerciale/marketing;
- La formalizzazione della Voice of the Customer in un documento che prende il nome di Specifica di prodotto;
- L'inserimento della famiglia di prodotto in un piano strategico di lungo termine, ovvero la Product Roadmap.

La specifica di prodotto non presenta soluzioni tecniche al suo interno, le quali verranno approfondite nella fase successiva del progetto, ma presenta la definizione dei requisiti di prodotto espressi in funzione dei bisogni del cliente ed altre informazioni ausiliarie come volumi di vendita previsti e target cost.

Tale documento inoltre non è rigido, ma può evolvere nel corso del progetto in funzione di quelli che sono i cambiamenti e le nuove informazioni che emergono nel corso dello sviluppo, in accordo al ciclo di miglioramento SPDCA.

Gate. I gate del nuovo processo di sviluppo hanno la funzione di fase di check sulla completezza e lo svolgimento delle attività richieste nel corso delle varie fasi del processo.

In particolare, il primo Gate prevede l'approvazione della Specifica di Prodotto del progetto contenuto nella Product Roadmap; tale approvazione sancisce formalmente il Kick-Off del progetto.

Questo Gate prevede un approccio Go/Kill in quanto, nel caso in cui la Specifica di prodotto non superi l'approvazione, non si procederà con l'avvio del progetto.

Fattibilità e Sviluppo Preliminare Alternative. La seconda macro-fase del processo di sviluppo nuovo prodotto prevede lo studio di fattibilità dei requisiti definiti all'interno della specifica di prodotto, con l'utilizzo di un documento di supporto denominato Specifica di Progetto. In particolare, l'obiettivo di questa seconda fase è quello di collegare i bisogni del cliente con le possibili modalità con le quali realizzare il prodotto. Lo studio di fattibilità consente di far emergere eventuali lacune all'interno della specifica di prodotto, e quindi la necessità di acquisire un maggior grado di conoscenza. Il tutto può portare alla modifica della specifica di prodotto stessa o alla definizione di "non fattibilità" del prodotto con i requisiti/target cost/budget definiti. Una delle possibili tecniche adottate in questa fase è il *Quality Function Deployment*, il quale permette di identificare e tradurre i bisogni e i desideri del cliente in requisiti e caratteristiche tecniche misurabili e di fissare i relativi livelli di performance del prodotto. L'identificazione di possibili soluzioni tecniche consente di sfruttare la conoscenza formalizzata e usufruibile generata dal processo di ricerca per la progettazione preliminare di alternative concorrenti. In parallelo, avviene la definizione preliminare del piano di marketing per i mercati target individuati.

Al termine della seconda macro-fase di sviluppo, il secondo Gate è focalizzato sull'approvazione di:

- Specifica di progetto;
- Team di sviluppo;
- Piano di sviluppo;
- Piano preliminare di marketing per ogni mercato target.

Sviluppo. Mentre il secondo stage del processo prevede uno sviluppo preliminare delle soluzioni alternative attraverso prototipazioni teoriche, esperimenti o modellazioni, nel terzo stage si ha lo sviluppo concreto delle alternative individuate e delle nuove alternative che possono emergere con l'acquisizione di ulteriore conoscenza, seguito da una serie di test con

l'obiettivo di filtrare le soluzioni più deboli. L'approccio adottato è quello del *Set-Based Concurrent Engineering*.⁴²

In particolare, alcune delle attività di sviluppo di questa terza fase prevedono:

- Sviluppo di formulati concorrenti;
- Sviluppo modalità di applicazione;
- Sviluppo tecniche produttive e set-up linee.

È importante sottolineare che tra le varie attività vi sono la progettazione concorrente del prodotto e del processo che dovrà realizzarlo in funzione di tutte le 4M+E del processo, coerentemente con uno dei punti cardine del *Lean Development*⁴³ che prevede di definire in fase di sviluppo l'intera Value Stream e non limitarsi alla sola progettazione del prodotto.

Il terzo Gate, che segue la fase di sviluppo, prevede tra le attività di check principali la formalizzazione della design review delle alternative progettuali ritenute valide in quel momento, della documentazione di marketing e delle etichette del prodotto. Vengono inoltre eseguite l'analisi dei costi industriali delle alternative e la selezione delle alternative con cui procedere con l'alfa test. L'approccio basato sulla progettazione di alternative concorrenti consente di ridurre il rischio che l'analisi al Gate provochi la sospensione del progetto, per la mancanza di un'alternativa valida, o massicce rilavorazioni.

Alfa Test. La fase di Alfa Test prevede l'esecuzione di una serie di test su un campione di prodotti per verificare che rispondano ai requisiti formalizzati nella specifica di prodotto (i quali possono essere evoluti rispetto a quelli contenuti nella specifica iniziale, grazie alla maggior mole di informazioni raccolte). Questa categoria di test viene svolta internamente all'azienda in un cantiere simulato.

Lancio produzione pre-serie e beta test. Nel caso in cui il campione di prodotto superi l'Alfa Test viene realizzata una pre-serie che verrà testata da clienti selezionati. La produzione della pre-serie avviene attraverso il test del processo produttivo. Il progetto di sviluppo termina con il lancio della produzione di massa e del piano di marketing e la formalizzazione della conoscenza acquisita.

⁴² Il *Set-Based Concurrent Engineering* (Progettazione per alternative concorrenti) è un approccio allo sviluppo che consiste nell'ideare, sviluppare e comunicare un set di soluzioni concettuali in parallelo. Il termine è stato formalizzato a metà degli anni '90 dai ricercatori statunitensi Sobek, Liker e Ward sulla base degli studi realizzati al processo di sviluppo prodotto di Toyota.

⁴³ Applicazione dei principi del Lean System al processo di sviluppo prodotto.

3.3 Organizzazione del processo

La mappatura dello stato futuro del processo di FILA, inserita nel paragrafo precedente, descrive le macro-attività necessarie per trasformare gli input al processo (come idee, bisogni del cliente, bisogni interni) in un prodotto realizzabile e, più in generale, nell'intera Value Stream necessaria alla realizzazione di tale prodotto.

L'approccio utilizzato per pianificare le attività del progetto di sviluppo sulla base di quanto descritto dalla mappatura dello stato futuro, è articolato su due livelli: livello di pianificazione strategico e livello di pianificazione esecutivo.

A livello strategico è adottato un diagramma di Gantt, il quale è uno strumento che consente di:

- Monitorare in modo visual l'avanzamento del progetto;
- Identificare le *milestones*;
- Pianificare il progetto con un orizzonte temporale di lungo termine.

All'interno del diagramma di Gantt vengono tempificate le macro-attività descritte nella FSM [Figura 17].

A livello di pianificazione esecutiva la fase di Lean Gap Analysis® del progetto di trasformazione lean ha fatto emergere la necessità di adottare un framework agile di supporto al team. Il bisogno di cambiare il precedente modo di fare è stato espresso all'interno di una sessione di workshop [Figura 17] direttamente dal personale coinvolto nel processo.

Con il team è emerso che l'attuale processo di sviluppo:

- Mette in difficoltà perché non è Visual;
- Non presenta un obiettivo condiviso e definito;
- Richiede il lavoro in team separati, sia spazialmente che dal punto di vista della sequenza delle attività;
- Porta a pianificare quello che in realtà non si conosce ancora bene.

I bisogni espressi dal team per l'organizzazione del nuovo processo sono i seguenti:

- Maggior precisione nel definire il carico di lavoro;
- Strumento visual per capire il carico di lavoro e l'avanzamento con un colpo d'occhio;
- Team di lavoro inter-funzionali;
- Definizione di cosa c'è da fare e del carico di lavoro da parte dell'owner dell'attività.

La soluzione che è si è deciso di adottare in modo da rispondere ai bisogni del team è l'implementazione del framework di pianificazione Scrum.

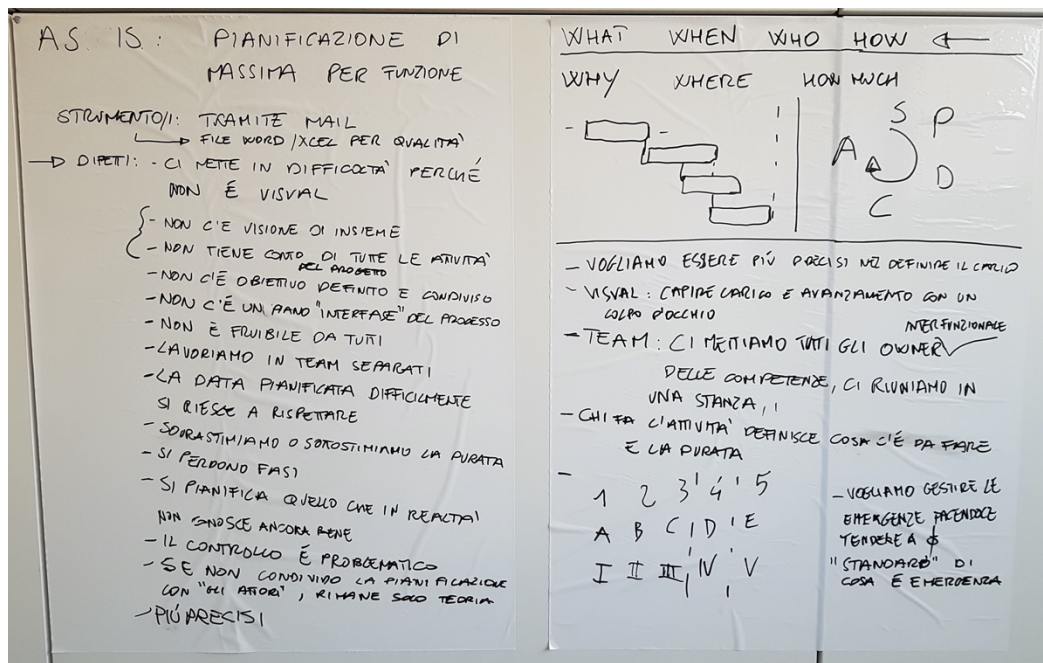


Figura 17 - Definizione linee guida del processo di pianificazione in FILA - Fonte: auxiell

La struttura che assume l'organizzazione dell'intero processo di Development appena descritta segue le logiche del modello Hybrid Stage - Gate esposte nel secondo capitolo dell'elaborato.

L'utilizzo del framework Scrum si inserisce all'interno del processo in seguito al primo Gate, quindi a partire dal secondo Stage. Come precedentemente descritto nel paragrafo 3.2.4.5, il Gate I sancisce formalmente l'avvio del progetto di sviluppo tramite l'approvazione della specifica di prodotto.

All'interno del secondo Stage è prevista la formazione del team di sviluppo. Una volta definito il team, i membri e il Product Owner possono riunirsi per sancire l'avvio della pianificazione del progetto con Scrum. In tale occasione, il team Scrum popola il Product Backlog con le informazioni contenute all'interno della specifica di prodotto, le quali rappresentano le conoscenze finora disponibili riguardo alle funzionalità del prodotto da sviluppare.

Nel complesso, è possibile rappresentare la modalità di gestione del processo di development di FILA come segue [Figura 18]:

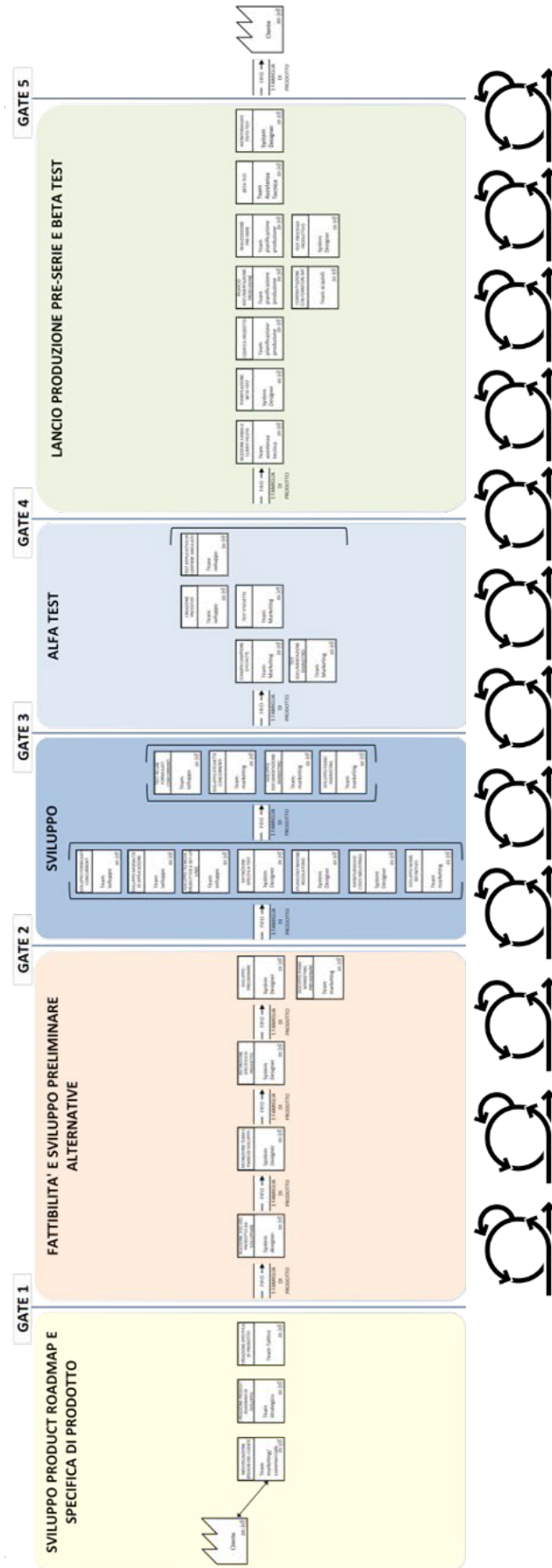


Figura 18 - Rappresentazione dell'organizzazione del processo di development in FILA

3.4 Formalizzazione degli standard di Scrum in FILA

Di seguito vengono descritte le componenti fondamentali che caratterizzano il framework e gli standard adottati in FILA per l'implementazione di Scrum nello sviluppo di prodotti professionali per la pulizia, mentre nel paragrafo 3.5 viene esposto un esempio di applicazione di tali standard.

3.4.1 I Ruoli

Team di sviluppo. Il team di Sviluppo in FILA è caratterizzato da un team ristretto formato da 4 persone presenti per l'intera durata del progetto e da altre figure introdotte nel team solo per alcune attività specifiche del progetto.

Il team ristretto di progetto è caratterizzato dalle seguenti competenze:

- Marketing Product Manager;
- Ricerca e Sviluppo;
- Assistenza tecnica;
- Operations e dell'industrializzazione.

Non è detto che le competenze sopra citate siano sufficienti per portare avanti l'intero progetto di sviluppo in quanto durante l'evento di raffinazione del Backlog può emergere il bisogno di coinvolgere una o più figure che non appartengono al team ristretto.

Analizzando nel dettaglio le storie del Backlog il team Scrum può rendersi conto che al suo interno non sono presenti tutte le competenze necessarie per garantire la multidisciplinarietà e l'autonomia nel portare avanti le attività. Per tale motivo viene introdotta un'ulteriore risorsa che le presenti, la quale è protagonista della raffinazione del Backlog, in quanto detiene le conoscenze necessarie per lo studio e l'analisi, e della successiva pianificazione, poiché si fa carico di portare avanti l'attività.

Il team Scrum in FILA è *autonomo*, possiede le competenze e le responsabilità per portare avanti le decisioni che prende in modo auto-organizzato e auto-gestito.

I membri del team di Scrum, oltre ad essere coinvolti nel progetto di sviluppo oggetto di applicazione del framework, devono svolgere altre mansioni all'interno dell'organizzazione, che impediscono la possibilità di dedicare completamente il team al progetto.

Il focus sull'eliminazione degli sprechi di dispersione e interruzione derivanti dal passaggio continuo da un progetto all'altro, porta il team a decidere di dedicare alle attività del progetto una finestra temporale ben definita all'interno della settimana lavorativa, lasciando un buffer temporale per attività al di fuori del progetto o ritardi nelle stesse.

Impostando una durata dello Sprint di due settimane, il team decide di dedicare 3 giorni della settimana nel portare avanti le attività di sviluppo definite in fase di Sprint Planning, le quali devono essere svolte in modo continuativo.

Infine, il team di sviluppo in FILA lavora in modo co-localizzato. In particolare, le cerimonie/eventi di Scrum vengono svolti in una *Obeya Room*⁴⁴ dedicata al team Scrum mentre le attività specifiche di sviluppo vengono portate avanti da ciascun membro in modo autonomo/di team in aree dedicate a seconda delle necessità, ma sempre all'interno dello stesso plant.

Scrum Master. Nella prima fase di introduzione del framework, il ruolo dello Scrum Master viene supportato dal consulente auxiell.

L'obiettivo del consulente come figura esterna all'organizzazione è duplice:

- Ricoprire il ruolo di *Coach* trasmettendo al team Scrum le regole e i comportamenti da adottare per implementare con successo il framework, supportando il team nella rimozione degli ostacoli che si presentano nel corso del progetto, siano essi interni o esterni ad esso;
- Formare uno Scrum Master autonomo che sia in grado di estendere il metodo all'interno dell'organizzazione.

Product Owner. La figura che ricopre il ruolo del Product Owner nel Team Scrum di FILA è definita attraverso una scelta presa in comune accordo dall'intero team in una specifica riunione di "investitura" nella cui fase iniziale sono state definiti gli specifici requisiti, mansioni e comportamenti che tale figura deve soddisfare (presentati nel dettaglio nel paragrafo 3.3.2).

⁴⁴ Il termine *Obeya* in giapponese significa "Grande Stanza". Il concetto è stato introdotto da Takeshi Uchiyamada, Chief Engineer della Prius, con l'obiettivo di riunire tutti i membri del team di sviluppo in una grande sala in cui era possibile controllare l'avanzamento del progetto e prendere le decisioni chiave per l'avanzamento di esso.

3.4.2 Gli Eventi

Di seguito viene analizzato nello specifico ciascun evento di Scrum che viene svolto in FILA, descrivendo gli standard adottati per l'implementazione e alcune considerazioni aggiuntive circa lo svolgimento dell'evento.

Backlog Refinement.

L'evento di raffinazione del Backlog viene svolto con una frequenza di almeno una volta per ogni Sprint, con una durata massima di 1h per ogni settimana di durata dello stesso.

Alla riunione vi partecipa l'intero Team Scrum.

Gli obiettivi di tale riunione sono illustrati in seguito:

- Rendere *Pronte* le storie del Product Backlog scelte per lo Sprint successivo. Raffinare le storie fattibili durante la durata di uno Sprint, ed eventualmente altre da tenere come buffer;
- Identificare le informazioni mancanti o le azioni a lungo *lead time* necessarie per rendere pronte le storie scelte;
- Pianificare le azioni necessarie a rendere *pronte* le storie scelte, entro la riunione di Sprint Planning successiva.

Le attività che vengono svolte dai vari membri del team sono le seguenti:

Il Product Owner:

- Sceglie le storie da raffinare in base alla priorità;
- Supporta il team Scrum a capire le storie.

Il Team Scrum:

- Studia le storie, definendo tutti i dettagli necessari per renderle *pronte* in base allo standard (che verrà approfondito in seguito);
- Se mancano informazioni, pianifica le azioni necessarie a reperirle, in modo tale da terminare la raffinazione entro la riunione di Sprint planning dello Sprint successivo.

Time keeper: tiene il tempo e fa presente quanto manca alla fine della riunione (È presente a ciascun evento di Scrum, ma non verrà più elencato in quanto svolge le stesse mansioni anche negli altri eventi).

Facilitatore: fa in modo che non si divaghi rispetto agli obiettivi della riunione (È presente a ciascun evento di Scrum, ma non verrà più elencato in quanto svolge le stesse mansioni anche negli altri eventi).

Storie da raffinare. Per inserire una storia da raffinare nel Backlog è necessario compilare un post-it descrivendo sinteticamente l'attività in modo che contenga le seguenti informazioni (illustrate in [Figura 19]):



Figura 19 - Standard FILA definizione storia da raffinare - Fonte: auxiell

Successivamente alla creazione di una storia risulta necessaria la sua prioritizzazione.

Lo standard adottato in FILA prevede l'utilizzo di un template [Figura 20] in cui viene visualizzata la *Matrice di sforzo-impatto* per supportare il team nella prioritizzazione della storia.

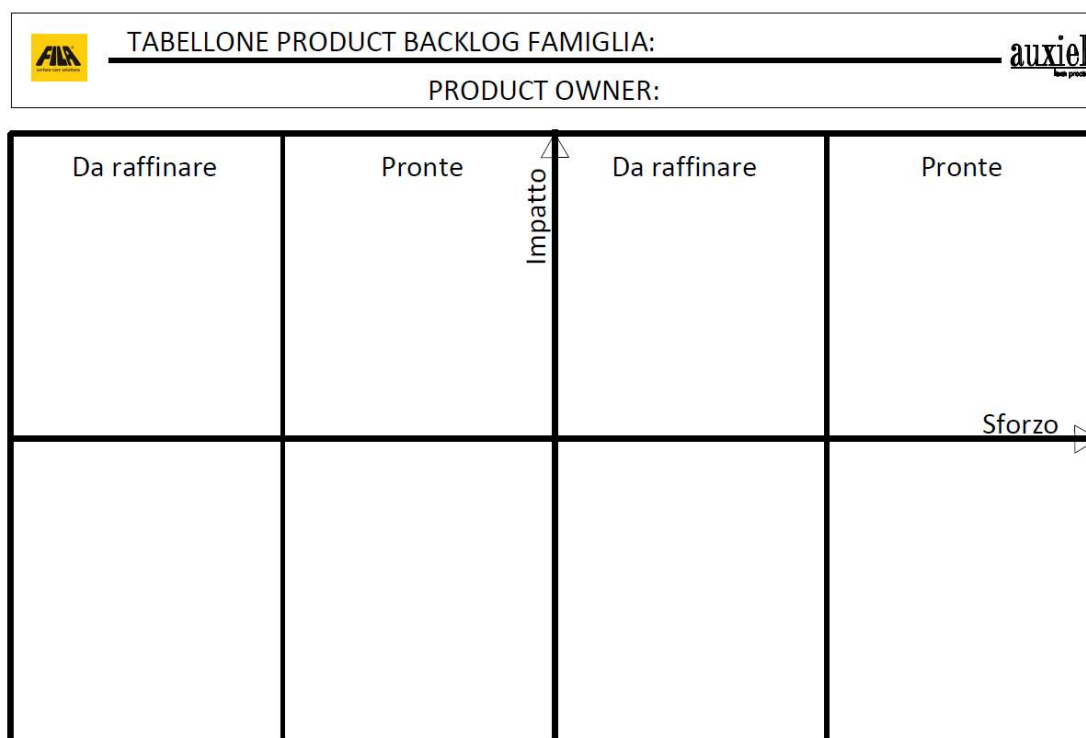


Figura 20 - Product Backlog - Fonte: auxiell

La *Matrice di impatto-sforzo* è il template adottato per rappresentare il Product Backlog.

Possiamo osservare che il Backlog presenta due sezioni per ciascun quadrante della matrice di sforzo-impatto: una per le *storie da raffinare* e una per le *storie pronte*.

Durante l'evento di raffinazione, la storia deve passare dalla sezione "da raffinare" alla sezione "pronta" del Backlog; non è però detto che venga posizionata nello stesso quadrante della matrice in quanto, analizzando nel dettaglio la storia, il team può rendersi conto che sia necessario assegnare una priorità differente rispetto a quella precedentemente assegnata.

L'utilizzo di tale strumento consente da un lato di avere un impatto immediato e visual delle storie che presentano una priorità maggiore e dall'altro di supportare la fase di prioritizzazione delle storie, fornendo al team due variabili dimensionali sulle quali basarsi.

Storie pronte. Come descritto precedentemente, il lavoro sulle storie non si limita alla sola creazione, ma anche alla loro raffinazione. Ciò si concretizza nello studio e analisi delle *storie da raffinare* in modo che siano chiari al team tutte le macro-attività che devono essere svolte (e il relativo carico di lavoro) per portarle a termine, l'obiettivo e a chi generano valore.

In seguito [Figura 21], viene riportato un esempio di *storia da raffinare* e di *storia pronta*:

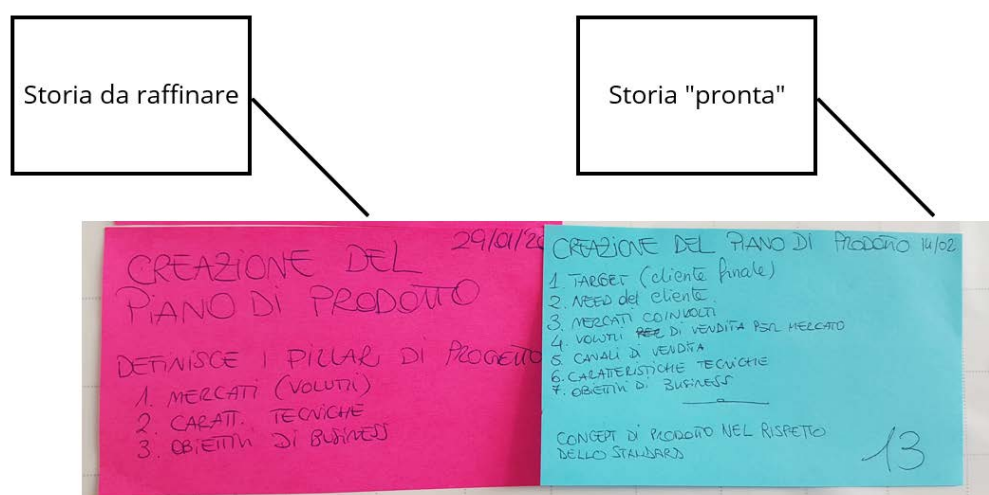


Figura 21 - esempio storia da raffinare e storia raffinata⁴⁵ - Fonte: auxiell

⁴⁵ L'esempio non segue lo standard: è stato utilizzato un file Excel per le storie raffinate, poiché lo spazio a disposizione nel post-it non era sufficiente. Per approfondimenti vedi paragrafo 3.5.

Lo standard attualmente utilizzato dal Team Scrum per la raffinazione delle storie, ovvero il processo di trasformazione della storia da raffinare a storia pronta, è il seguente [Figura 22]:

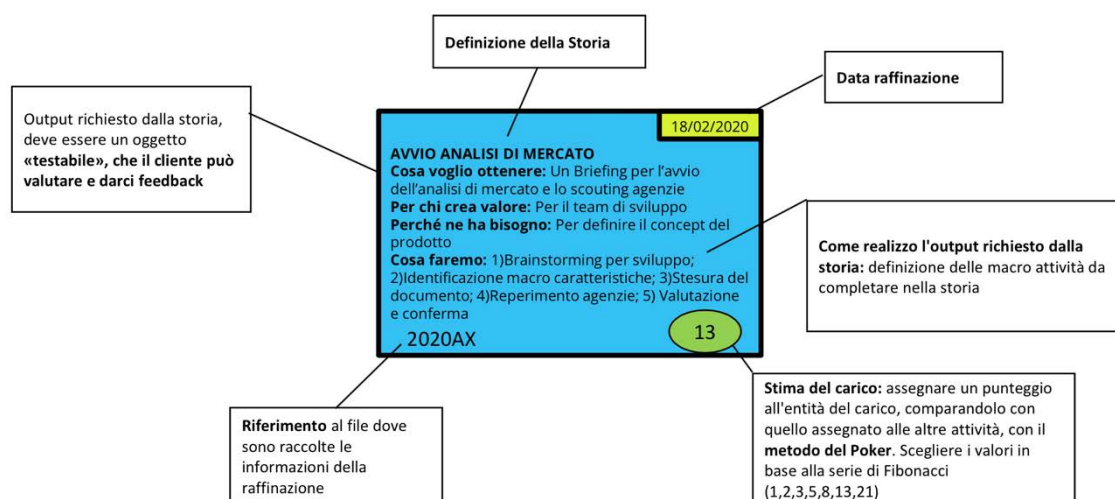


Figura 22 - standard FILA storia pronta - Fonte: auxiell

In particolare, per la definizione della *storia pronta* il team deve formalizzare i seguenti aspetti:

- *Cosa voglio ottenere*: l'output della storia, un oggetto “testabile” sul quale il cliente (sia esso interno o esterno) possa generare delle valutazioni e dare feedback per supportare l'apprendimento del team;
- *Per chi crea valore*: ciascuna attività che viene svolta nel processo di sviluppo deve essere focalizzata a generare valore per il cliente, sia esso interno o esterno all'organizzazione; per tale motivo è importante che il team abbia ben chiaro a chi genera valore l'output della storia, in modo da cercare di massimizzarlo, oppure rendersi conto che l'attività non genera valore (quindi è spreco) e cercare di minimizzarlo;
- *Perché ne ha bisogno*: il team viene focalizzato sull'obiettivo da raggiungere attraverso il completamento di tale storia. La riflessione sul *perché* dell'attività supporta la generazione di idee e crea chiarezza sulle attività con le quali portarla a termine;
- *Come realizzo l'output richiesto dalla storia*: le macro-attività necessarie per portare a termine la storia;
- *Stima del carico richiesto per portare a termine la storia*: il carico viene definito assegnando un dimensionamento relativo⁴⁶ a ciascuna storia,

⁴⁶ La stima del carico di lavoro con dimensionamento relativo consiste nella comparazione dimensionale tra due compiti. Ad esempio, se il team ritiene che una storia del Backlog presenta un carico di lavoro maggiore rispetto ad un'altra, allora a tale storia dovrà essere assegnato un punteggio superiore della serie di Fibonacci. Se una storia possiede un punteggio molto elevato rispetto alle altre, può essere un alert del fatto che tale storia non può essere completata nella durata di uno Sprint e quindi deve essere suddivisa in più storie con un punteggio inferiore.

sfruttando il *metodo del Poker*. Il metodo del Poker prevede che in fase di raffinazione ciascun membro del team di sviluppo assegni alla storia un valore appartenente alla serie di Fibonacci⁴⁷, indicativo del carico di lavoro che il membro del team ritiene che la storia richieda. Il valore viene scritto su un foglietto di carta senza farlo vedere agli altri membri; successivamente tutti i punteggi vengono scoperti contemporaneamente, in modo da evitare qualsiasi forma di condizionamento tra i vari membri del team. Se l'assegnazione del punteggio non dovesse generare il consenso dell'intero team, seguirebbe una breve sessione di confronto in cui si cercherebbero di chiarire i motivi dell'assegnazione da coloro che hanno definito il punteggio superiore e inferiore. L'obiettivo del confronto è quello di trovare un accordo tra i vari membri motivando le proprie considerazioni con dei ragionamenti, in modo da far raggiungere il consenso al team o optare per l'assegnazione della media delle stime. Nel caso in cui non si raggiunga una visione comune, è responsabilità dello Scrum Master mediare verso il consenso.

⁴⁷ La Sequenza di Fibonacci è un andamento numerico in cui il numero successivo è la somma dei due che lo precedono, per esempio: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55. L'utilizzo della serie di Fibonacci è supportato dal fatto che i numeri che la compongono sono abbastanza distanti da permetterci di cogliere immediatamente la differenza tra di essi e, di conseguenza, è facile schierarsi da una parte o dall'altra.

Sprint Planning.

L'evento di pianificazione viene svolto all'inizio di ciascuno Sprint, con una durata massima di un'ora per ogni settimana di durata dello stesso.

I partecipanti alla riunione sono sia il Team Scrum che il Product Owner.

Gli obiettivi della riunione sono i seguenti:

- Definire e concordare lo *Sprint Goal*;
- Definire il punteggio target da raggiungere nello Sprint, ovvero la somma dei punteggi delle storie che vengono selezionate, usando come riferimento i punteggi degli Sprint precedenti;
- Scegliere le storie da fare per raggiungere lo Sprint Goal;
- Studiare come raggiungere l'obiettivo previsto nelle storie;
- Rivedere e concordare il punteggio assegnato alle storie;
- Determinare e pianificare i task per i primi giorni dello Sprint.

Lo standard di pianificazione dello Sprint prevede l'assegnazione delle seguenti attività:

Il *Product Owner* è responsabile di proporre lo Sprint Goal e le storie su cui lavorare per raggiungere lo Sprint Goal. Inoltre, deve supportare il Team di Sviluppo a capire le storie, ovvero il *come*.

Le storie selezionate in fase di pianificazione sono scelte in base alla loro priorità, ovvero in funzione della loro posizione rispetto alla matrice di impatto-sforzo. In particolare, la scelta del Product Owner deve seguire la seguente logica (illustrata nella [Figura 23]): in un primo momento dovranno essere realizzate le storie che permettono di ottenere un elevato impatto, o che possiedono una priorità elevata, con un basso sforzo; le storie con un basso impatto e un basso sforzo dovranno essere realizzate solamente se effettivamente necessarie per il raggiungimento dell'obiettivo. Le storie con elevato impatto ed elevato sforzo dovranno essere valutate e pianificate con una maggior accuratezza, in quanto richiedono un effort elevato tale da doverci dedicare uno sprint o richiedere risorse/competenze aggiuntive. Infine, le storie che comportano un basso impatto a fronte di un elevato sforzo possono essere posticipate o addirittura non fatte nel caso in cui ci si accorga che non generano sufficiente valore.

Il Development Team:

- Discute la proposta di Sprint Goal e concorda con il Product Owner le storie da includere all'interno dello Sprint;
- Studia come raggiungere l'obiettivo previsto nelle storie;
- Espone le storie in task della durata non superiore alla giornata;
- Pianifica i task per i primi giorni dello Sprint.

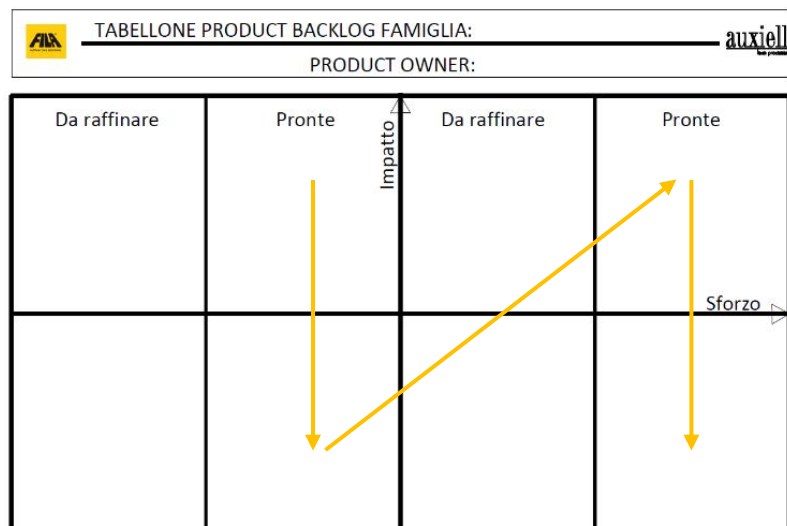


Figura 23 - Sequenza di selezione delle storie dal Backlog

Gli strumenti che vengono utilizzati durante la pianificazione dello Sprint, oltre al Product Backlog che è stato illustrato nella raffinazione del Backlog, sono i seguenti:

Tabellone Sprint. Una volta selezionate le storie da portare a termine entro la durata dello Sprint in corso, i relativi post-it vengono spostati all'interno della colonna "Da Fare" del tabellone Sprint, il quale rappresenta lo Sprint Backlog.

Successivamente, durante le riunioni di Daily Scrum viene monitorato l'avanzamento delle storie, le quali possono essere spostate nella colonna "In corso" quando sono stati presi in carico i relativi task che concorrono al loro completamento, e nella colonna "Fatte" quando tutti i relativi task sono stati completati e il risultato raggiunto è coerente con lo standard di *Storia Fatta*.

Di seguito viene riportato il template adottato in FILA per la visualizzazione dello Sprint Backlog [Figura 24].

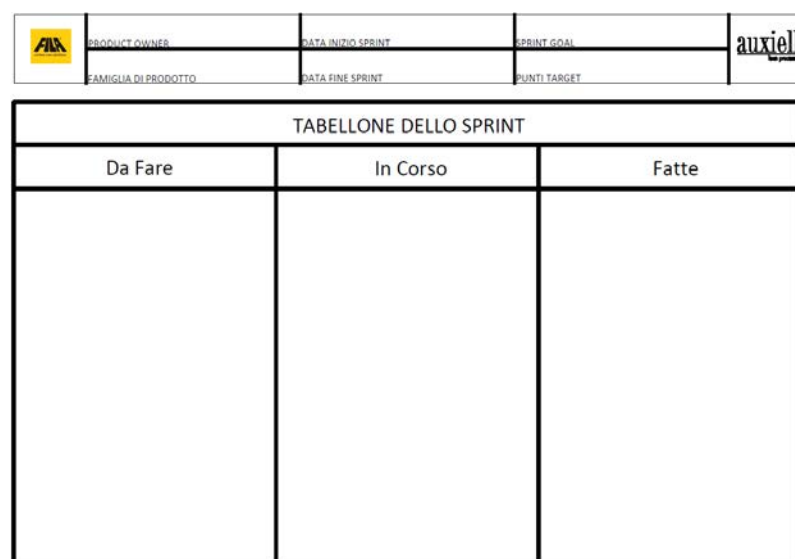


Figura 24 - Tabellone dello Sprint - Fonte: auxiell

Tabellone dei task. Il completamento di ciascuna storia presente all'interno dello Sprint Backlog prevede la pianificazione e lo svolgimento di una o più attività, tipicamente della durata di un giorno, che i membri del team si autoassegnano.

Per tale motivo viene utilizzato uno strumento di supporto parallelo al Tabellone Sprint, ovvero il *tabellone dei task*. All'interno di tale tabellone [Figura 25] vengono inseriti i task, sotto forma di post-it, che ciascun membro del team deve svolgere in modo da portare a termine le storie che sono state pianificate durante l'evento di Sprint Planning.

Ciascun membro del Team è responsabile di pianificare le proprie attività, ovvero di definire un determinato giorno dello Sprint in cui svolgerà l'attività, e di portare a termine le attività entro la data definita.



The image shows a template for a 'Tabellone dei Task' (Task Board). It features a header bar with a yellow logo on the left, the title 'TABELLONE DEI TASK' in the center, and the 'auxiell' logo on the right. Below the header is a grid with 7 columns and 6 rows. The first column is labeled 'Team members' and the subsequent columns are labeled with days of the week: 'Lun', 'Mar', 'Mer', 'Gio', 'Ven', and 'Next'. The grid is designed for team members to assign tasks to specific days.

Team members	Lun	Mar	Mer	Gio	Ven	Next

Figura 25 - Tabellone dei Task - Fonte: auxiell

Daily Scrum.

La riunione di Daily Scrum viene svolta giornalmente dallo Scrum Team di fronte ai tabelloni dello Sprint e si svolge con una durata massima di quindici minuti.

Questo evento giornaliero rappresenta una fase di *check* intermedia del lavoro del team e deve raggiungere gli obiettivi esposti in seguito:

- Ciascun membro del team risponde alle seguenti domande:
 - Ho completato i task previsti per ieri? [SI/NO]
 - Quali problemi mi hanno rallentato?
 - Come posso risolvere questi problemi e recuperare l'eventuale ritardo?
 - Cosa farò oggi?
- Il team aggiorna il tabellone Sprint monitorandone l'avanzamento delle storie e, nel caso in cui siano presenti, certifica le *storie fatte*.

Le attività definite all'interno dello standard di svolgimento dell'evento di Daily Scrum per i diversi ruoli sono le seguenti:

Ciascun membro del Team di Sviluppo:

- Annuncia i task completati ed elimina i post-it dal tabellone dei task;
- Supportato dal team, individua problemi e azioni correttive, con focus sul recupero di eventuale ritardo;
- Pianifica i propri task per la giornata, condividendoli con il team, compilando in prima persona i post-it.

Product Owner:

- Monitora l'avanzamento delle attività verso il raggiungimento dello Sprint Goal;
- Collabora con il team ad analizzare problemi e a pianificare azioni correttive;
- In caso di anticipo sul piano, sceglie storie dal buffer di quelle pronte, da aggiungere nello Sprint in corso;
- Fornisce eventuali ulteriori chiarimenti sulle storie.

Sprint Review.

Al termine di ciascuno Sprint viene svolta la riunione di Sprint Review che funge da fase di *check* e *act* dello Sprint stesso e ha una durata di un'ora per ciascuna settimana di svolgimento.

Gli obiettivi di tale riunione sono i seguenti:

- Presentare le storie fatte/non fatte e come il lavoro è stato svolto, in modo da accertarsi che l'output sia conforme a quanto richiesto dagli stakeholder;
- Aggiornare il Product Backlog;
- Discutere dei problemi incontrati, delle soluzioni individuate e pianificare relativa standardizzazione/estensione a casi simili;
- Analizzare le cause di eventuali scostamenti tra punteggio effettivo e punteggio target dello Sprint, pianificando azioni correttive/migliorative, le quali vengono scritte come storie e inserite nel Backlog.

Durante la riunione di Sprint Review, il Product Owner:

- Esamina e presenta le storie fatte e quelle non fatte, aggiorna il *grafico storico di velocità degli Sprint*;
- Aggiorna il Product Backlog e stima le date di consegna delle attività rimanenti;
- Pianifica la standardizzazione delle soluzioni individuate e l'estensione ad altri casi simili;
- Guida l'analisi di eventuali scostamenti tra il *punteggio target* dello Sprint e quello *effettivo*. Nel caso in cui alcune storie non siano state completate (e quindi vi sia uno scostamento negativo tra il punteggio target e quello raggiunto), viene sostenuta una breve sessione di *problem solving* dove il team è spinto, con il supporto dello Scrum Master, alla ricerca della causa radice di tali criticità; successivamente vengono pianificate delle attività per la risoluzione dei problemi individuati, con l'obiettivo di aumentare progressivamente la velocità del team.

Per quanto riguarda il Team di Sviluppo:

- Spiega come il lavoro è stato fatto;
- Passa in rassegna ciò che è andato bene e cosa no, i problemi incontrati e come sono stati risolti;
- Collabora con il Product Owner in tutte le attività previste.

L'analisi dell'andamento delle storie completate dal team nel corso dei vari Sprint è supportata da uno strumento visual, ovvero il *grafico storico di velocità degli Sprint* (di cui è riportato il template [Figura 26]), il quale consente di capire da un lato se il team ha acquisito consapevolezza rispetto al carico di lavoro che è in grado portare a termine rispetto a quanto pianificato e dall'altro se il team ha aumentato la propria velocità, fungendo quindi da indicatore della quantità di ostacoli eliminati nel processo di sviluppo.

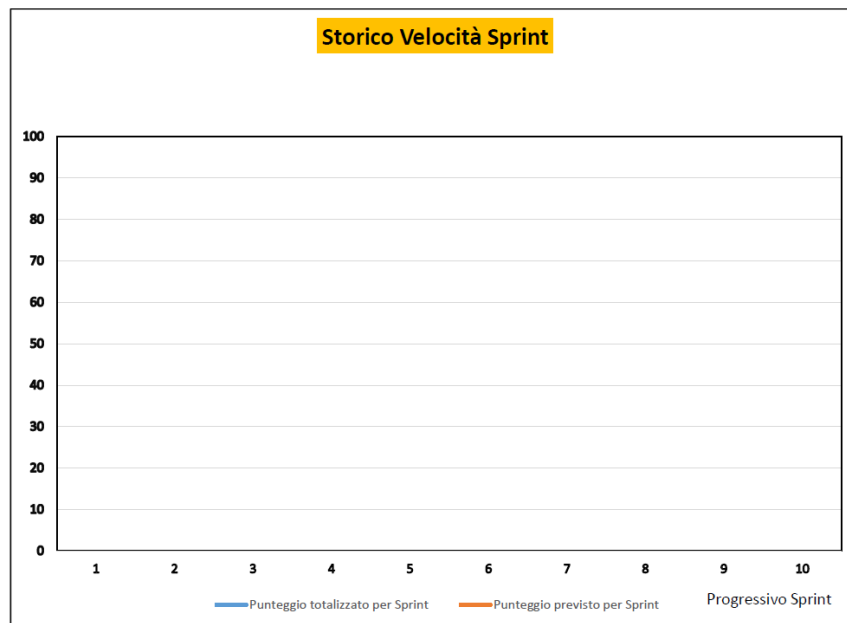


Figura 26 - Grafico storico di velocità degli Sprint - Fonte: auxiell

3.5 Esempio di svolgimento di uno Sprint in FILA

Come descritto nel paragrafo 3.3, l'organizzazione del processo in ottica Hybrid Stage-Gate impostata in FILA prevede l'utilizzo del framework di pianificazione Scrum a partire dal kick-off del progetto, che coincide con il primo Gate.

Lo specifico progetto di sviluppo in cui è inserito lo Sprint che verrà descritto in seguito è stato pre-approvato dalla direzione, in quanto ritenuto strategico per l'azienda sulla base dei dati a disposizione.

Assieme al team e al management si è deciso di procedere comunque allo svolgimento del primo stage come fase all'interno del progetto, con un triplice obiettivo:

- Creare gli standard di processo per il primo stage, dall'acquisizione della VoC fino alla definizione della specifica di prodotto e il suo inserimento nella Product Roadmap;
- Utilizzare gli standard creati per verificare attraverso la ricerca di mercato se le idee di sviluppo sono valide;
- Trasmettere al team le regole di Scrum.

In questa prima fase del progetto si è definito assieme al team di svolgere Sprint con una durata fissa di due settimane. All'interno dello Sprint i membri del team sono dedicati alle attività del progetto per un numero di giornate prestabilito, nello specifico nel primo stage si è deciso di dedicare due giornate, che però potranno aumentare negli Stage successivi.

Durante il primo meeting, il team si è occupato di definire le storie da portare a termine nella prima parte del progetto.

Vengono a titolo esemplificativo riportate tre storie che sono state inserite all'interno del Backlog [Figura 27]:

07/06/2020	07/06/2020	07/06/2020
AVVIO ANALISI DI MERCATO	CONFERMA ANALISI DI MERCATO	ELABORAZIONE DATI ANALISI DI MERCATO
Identificare target, need e posizionamento prodotti	Dare avvio all'analisi da parte dell'agenzia	Valutare l'output dell'analisi

Figura 27 - Storie compilate - Fonte: auxiell

In questo primo meeting la principale difficoltà manifestatasi al team è stata quella di dover ragionare per obiettivi. La tendenza era quella di pianificare delle attività invece che ragionare per deliverable. Il team è stato supportato nel migliorare il proprio modo di scrivere le storie da inserire all'interno del Backlog, facendo in modo che in fase di Review queste siano facilmente testabili e che quindi il Product Owner possa facilmente identificarne il completamento.

In seguito alla compilazione delle storie, il team ha proseguito con il loro inserimento all'interno del Backlog, coerentemente con gli standard definiti. La valutazione dell'impatto delle storie è principalmente legata ai vincoli di precedenza che le collegano.

L'output ottenuto è rappresentato in [Figura 28]⁴⁸:

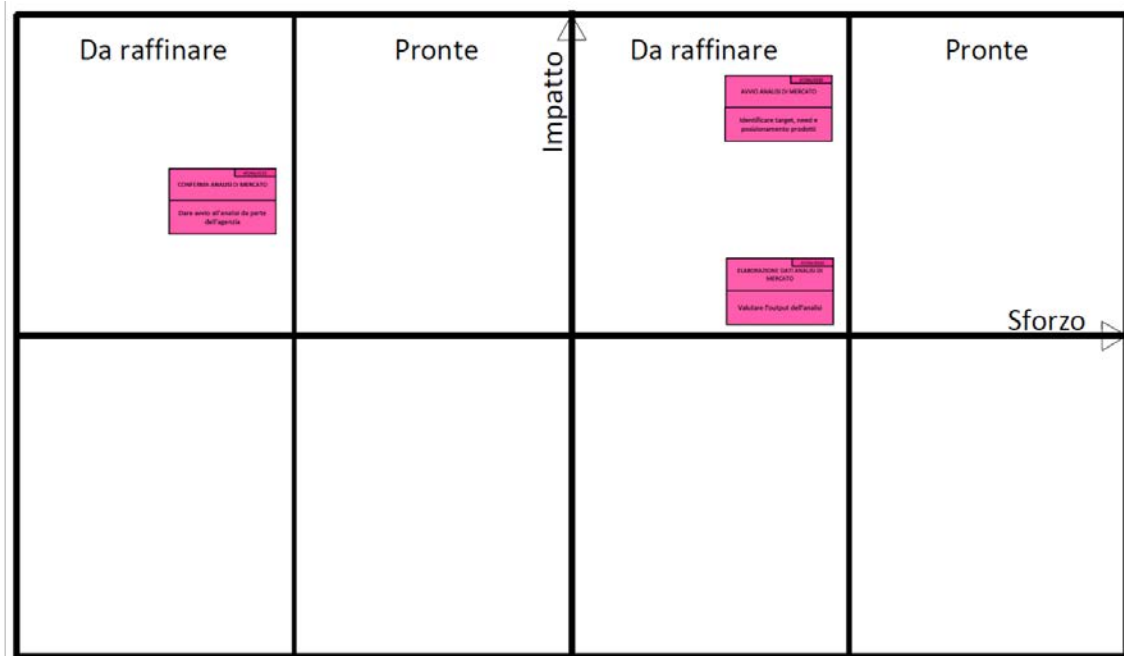


Figura 28 - Backlog virtuale con le storie da raffinare - Fonte: auxiell

In seguito al meeting dedicato alla creazione delle storie del Backlog, il team si è riunito per l'evento di raffinazione. In questa riunione si è optato per trasferire le informazioni dai post-it, che risultavano uno spazio troppo piccolo per raccogliere tutte le informazioni definite nello standard, ad un foglio Excel condiviso.

Le informazioni contenute all'interno dell'Excel rispecchiano lo standard di raffinazione delle storie illustrato al paragrafo 3.4.2. In [Figura 29] è riportato come esempio il contenuto della raffinazione della storia *Avvio analisi di mercato*.

Peso	n	Attività	Owner	Gli output della storia	Caratteristiche di ciascun output	Perché?	Come?
13	1	Definizione del Brief	Sara, Maria, Nicola, Paolo, Luca	- identificazione del prodotto (CARTA D'IDENTITÀ) - definizione del/dei target - identificazione canale - identificazione dei mercati - requisiti del cliente	- categoria, funzione d'uso, caratteristiche chimiche, delta resa, posizionamento prezzo, eco-sostenibilità - tipologia di consumatore/i target fascia d'età e genere - scelta dei canali più adatti (canali on-line, off-line, diretto) - scelta dei mercati più adatti (regolatorio, assistenza, richiesta dal cliente, supply chain) - lista dei bisogni dei clienti espressi e inespresi	- per completare la descrizione del prodotto per il brief - dimensionare il progetto per il carico/complessità - identificare i need del consumatore - indirizzare lo sviluppo R&D e la scelta della soluzione più idonea - per essere sicuri di raggiungere il target - individuare i mercati con maggiore potenzialità - per soddisfare il cliente	- brainstorming - compilazione di un documento standard - 4M+ E: Man: Commerciale di riferimento + team (CEO's per approvazione) - Machine: pc - Material: no - Method: documento standard di carta d'identità - Enviroment: OBEYA ROOM

Figura 29 - Storia raffinata - Fonte: auxiell

⁴⁸ È stato riportato un Backlog virtuale in quanto non è stata raccolta reportistica fotografica di questo specifico passaggio.

Una prima difficoltà riscontrata in questa fase è legata all'assegnazione del punteggio alla storia, in quanto è atteggiamento comune quello di ragionare in termini di ore da investire per portare a termine l'attività, la maggior parte delle volte non riuscendoci però in modo efficace. Il team è stato supportato ad assegnare il valore della serie di Fibonacci confrontando lo sforzo richiesto dai diversi deliverable definiti nelle storie.

Una seconda difficoltà riscontrata è stata quella di dover ragionare distinguendo tra il deliverable concreto da raggiungere e i requisiti dello stesso, in modo da determinare una macro-sequenza di attività e le varie componenti delle 4M+E, oltre che per determinare il carico richiesto dal deliverable. Questa seconda difficoltà è stata superata definendo assieme al team uno standard di raffinazione che li supportasse durante l'evento [Figura 22-29].

Al termine della riunione di raffinazione del Backlog è seguito l'evento di Sprint Planning durante il quale il team, supportato dal Product Owner, ha pianificato tre storie per lo Sprint in corso:

- Avvio analisi di mercato, con un punteggio di tredici punti;
- Identificazione Macro del prodotto, con un punteggio di cinque punti;
- Conferma dell'analisi di mercato, con un punteggio di cinque punti;

In totale, il punteggio target che il team si è prefissato di raggiungere è di ventitré punti storia.

L'aspetto che assume il Backlog in seguito alla riunione di Sprint Planning è il seguente [Figura 30]:

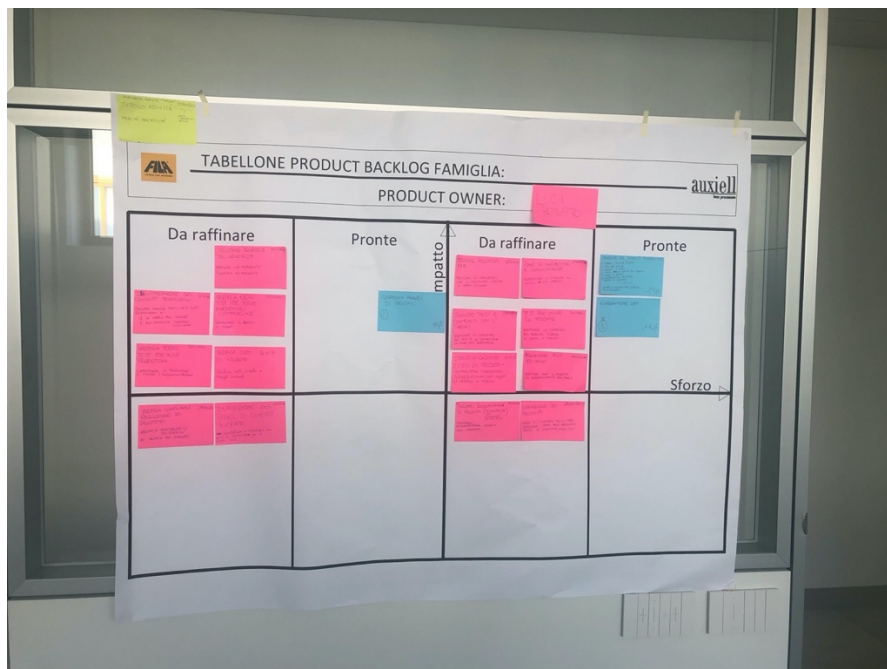


Figura 30 - Product Backlog in FILA

Con i post-it di colore rosa sono definite le storie ancora da raffinare, mentre con i post-it azzurri sono rappresentate le storie pronte.

Le storie pianificate per lo Sprint in corso sono inserite all'interno del Tabellone dello Sprint [Figura 31]:

Figura 31 - Tabellone dello Sprint del progetto in FILA

Dopo aver pianificato le storie da completare nell'arco dello Sprint, il team ha pianificato i task correlati per le prime due giornate e successivamente li ha inseriti all'interno del Tabellone dei Task sotto forma di post-it [Figura 32]. In questa fase è risultato fondamentale sensibilizzare il team alla stesura di task che risultassero completabili nell'arco della giornata lavorativa, altrimenti lo standard prevede la suddivisione del task in due sotto-task. Questa scelta è legata al fatto che durante le riunioni di Daily Scrum il team è in grado di ispezionare eventuali deviazioni al processo e adattarsi tempestivamente, trovando contromisure e rimuovendo eventuali ostacoli.

Alcuni dei Task pianificati sono:

- Definire il posizionamento di prezzo;
- Definire l'eco-sostenibilità del prodotto;
- Definire la funzionalità d'uso;
- Definire le caratteristiche chimiche macro del prodotto;

In questo evento sono emerse alcune criticità già descritte nell'analisi di letteratura nell'utilizzo di Scrum nello sviluppo di prodotti hardware, ovvero la presenza di competenze verticali che rendono difficile la pianificazione in alcune fasi del progetto.

Tale criticità è emersa anche durante una delle riunioni di Daily Scrum.

dello Sprint. Gli obiettivi raggiunti sono “identificazione delle macro del prodotto” e “avvio dell’analisi di mercato”, ma non è stata completata la storia “conferma dell’analisi di mercato”, che prevedeva l’invio delle informazioni raccolte all’agenzia identificata.

I principali ostacoli identificati dal team sono coerenti con quanto emerso durante le riunioni di Daily Scrum, ovvero una complessità diffusa nel dedicare la totalità del tempo pianificato allo svolgimento delle attività del progetto.

Il supporto iniziale al team per la definizione delle storie pronte ha consentito al Product Owner di identificare con facilità se la storia analizzata poteva ritenersi completata o se vi erano ancora delle componenti mancanti.

L’analisi del caso studio appena discusso è riportata all’interno dell’ultimo capitolo dell’elaborato assieme ad un confronto con un ulteriore caso studio, presente nel capitolo che segue.

Capitolo 4.

Il Caso studio: FreTor s.r.l.

*“Non avere problemi
è il più grande dei problemi”*

Taiichi Ohno

Nel seguente capitolo è esposto il modello Hybrid Stage - Gate adottato nel processo di development in FreTor s.r.l., società che ho avuto il piacere di visitare e grazie alla quale ho avuto l'opportunità di osservare la metodologia in analisi applicata ad una realtà differente. Nella parte finale del capitolo è riportato un esempio di applicazione del framework Scrum all'interno di uno dei progetti dell'azienda.

4.1. Introduzione a FreTor s.r.l.

“Serve la migliore tecnologia per migliorare il futuro.”⁴⁹

“Mettiamo insieme esperienza e tecnologie meccaniche, elettriche e software per aumentare il valore per il nostro Cliente, riducendo tempi e costi di produzione ed incrementando la qualità del prodotto. Liberiamo gli operatori dalle attività ripetitive a basso valore aggiunto e li aiutiamo a riqualificarsi portando creatività nel proprio lavoro.”⁵⁰



Figura 33 - Logo FreTor s.r.l. - Fonte: <https://www.fretor.com>

FreTor s.r.l.⁵¹ nasce nel 1976 come azienda familiare il cui core business è focalizzato sulla realizzazione di particolari meccanici di precisione attraverso l'utilizzo di fresa e tornio, da cui ne deriva il nome.

Nel corso degli anni l'azienda incrementa il proprio know-how interno arrivando a realizzare nel 1996 la prima macchina industriale completamente automatica per il carico e scarico di piccoli oggetti.

Oggi FreTor rappresenta una realtà affermata nelle automazioni industriali, progettazione robotica e meccanica di precisione per i settori farmaceutico, automotive, medicale, aerospaziale, alimentare e occhialeria.

Nel 2018 entra a far parte del Gruppo Brovedani, società multinazionale leader nella meccanica di precisione nel settore automotive.

Nel 2019 FreTor avvia un progetto di Complete Lean Transformation con l'obiettivo di migliorare i propri processi, supportata da auxiell.

⁴⁹ Vision di FreTor. Fonte: <https://www.fretor.com/vision/>

⁵⁰ Mission di FreTor. Fonte: <https://www.fretor.com/mission/>

⁵¹ D'ora in poi all'interno dell'elaborato verrà usato solamente FreTor.

4.2. Organizzazione del processo

Il modello di gestione adottato da FreTor per la gestione dei propri processi è riconducibile alla tipologia Engineer to Order, sfruttando la classificazione definita da Wortmann⁵². In particolare, questa tipologia di sistemi è caratterizzata da un progetto di sviluppo che comprende sia la fase di progettazione che quella di produzione e non inizia finché non vi è l'acquisizione dell'ordine del cliente.

All'interno della fase di Lean Gap Analysis⁵³ del progetto di trasformazione di FreTor è stata definita la Future State Map che dovrà assumere il processo.

A livello macro il processo è organizzato attraverso un modello Stage-Gate caratterizzato da 3 macro-fasi:

- Il primo Stage prevede l'analisi di fattibilità delle richieste del cliente, la definizione della specifica di prodotto sulla base di tali richieste e l'elaborazione dell'offerta, che prevede anche la definizione di un diagramma di Gantt del progetto;
- Il primo Gate è caratterizzato da un approccio Go/Kill e il suo superamento è legato all'accettazione dell'offerta da parte del cliente, la quale sancisce formalmente il kick-off del progetto;
- Durante il secondo Stage viene incaricato un Project Manager e si dà avvio alla fase di progettazione assegnando un team dedicato al progetto, che si occupa di:
 - Progettazione meccanica;
 - Progettazione software;
 - Progettazione pneumatica;
 - Progettazione elettrica;
 - Acquisti;
- Il secondo Gate rappresenta un momento di check rispetto al completamento delle attività necessarie al passaggio all'ultima fase del processo, quali la creazione e la stampa dei disegni, la documentazione per la manualistica e la formalizzazione della conoscenza condivisa;
- Nel corso del terzo Stage avviene l'approvvigionamento, la produzione, l'assemblaggio, il collaudo e infine l'installazione presso il cliente;
- L'ultimo Gate rappresenta sia la fase di check dell'ultimo Stage, con la verifica della creazione della check list di sicurezza e della certificazione macchine, sia la fase di *act* dell'intero progetto con la formalizzazione della conoscenza acquisita.

⁵² La classificazione di Wortmann suddivide gli impianti in funzione del customer decoupling point, ovvero secondo il momento in cui la produzione passa da essere su previsione ad essere basata sull'ordine dei clienti. Fonte: https://it.wikipedia.org/wiki/Sistema_di_produzione

⁵³ Per una descrizione più approfondita dello svolgimento della fase vedere paragrafo 3.2.4.

A livello operativo, le attività del progetto vengono gestite utilizzando l'approccio agile garantito dal framework Scrum.

Nello specifico, la pianificazione con Scrum ha inizio con il kick-off del progetto all'inizio del secondo Stage, in quanto avvengono:

- Assegnazione del Project Manager, ad opera del Program Manager;
- Assegnazione dei progettisti, ad opera dei responsabili di progettazione;
- Approvazione della specifica di prodotto, la quale rappresenta l'input alla generazione del Backlog.

La pianificazione tramite l'utilizzo di Scrum continua anche nella fase di produzione e montaggio e termina con la fine del progetto, che coincide con l'installazione in loco dal cliente e il meeting di fine commessa.

Nel complesso, la modalità di gestione del processo di development in FreTor segue le logiche dell'approccio Hybrid Stage-Gate [Figura 34].

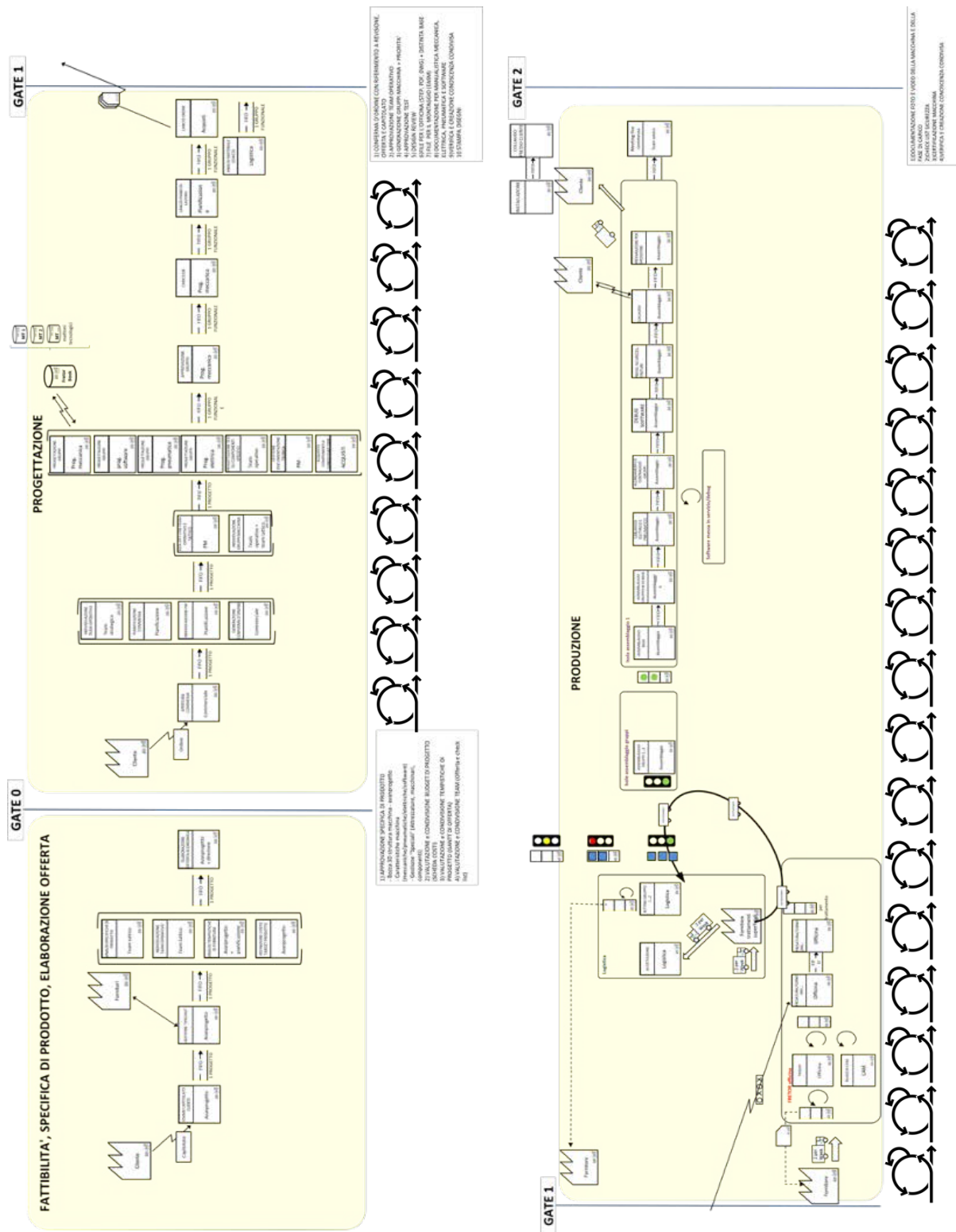


Figura 34 - Rappresentazione Future State Map FreTor con aggiunta del dettaglio degli Sprint- Fonte: auxiell

4.2.1 Attività preparatorie alla pianificazione con Scrum

Come descritto all'interno del primo capitolo dell'elaborato, uno dei requisiti fondamentali per il corretto utilizzo di Scrum è la creazione e l'aggiornamento del Backlog del prodotto, il quale rappresenta uno degli artefatti del framework⁵⁴.

In un contesto organizzativo caratterizzato da un modello di gestione ETO, il progetto di sviluppo ha avvio solamente in seguito ad una richiesta formale da parte del cliente. In FreTor, tale richiesta formale del cliente è racchiusa in un documento denominato *capitolato* che, a sua volta, viene integrato nella *specifica di prodotto*, la quale viene approvata formalmente durante il primo Gate del processo.

Con l'obiettivo di implementare i principi di trasparenza ed ispezione, il team formalizza le richieste generate dal cliente, ovvero i *requisiti*, in un apposito strumento denominato *tabellone dei requisiti*. Ciò avviene in una riunione dedicata all'inizio del secondo stage, la cui durata può essere variabile in funzione della complessità della commessa.

Lo standard di inserimento del requisito nel tabellone [Figura 35] prevede che esso debba essere collegato alla richiesta del cliente presente all'interno del capitolato, debba produrre valore per il cliente e il suo output debba essere testabile, in modo che possa essere chiaro e condiviso se il requisito può essere considerato "fatto".

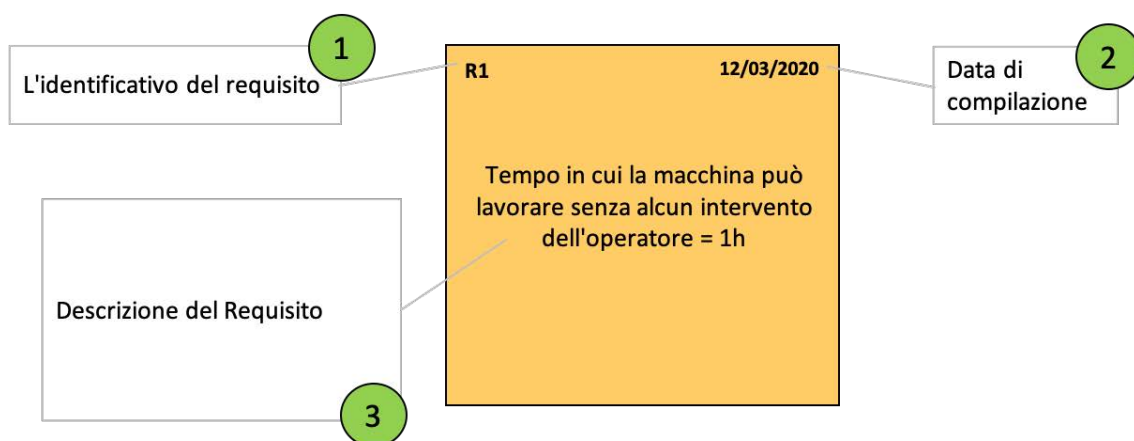


Figura 35 - Standard compilazione requisito - Fonte: auxiell

La formalizzazione dei requisiti consente al team di ragionare riguardo alla chiarezza di essi, in quanto non è detto che il cliente abbia le idee chiare su tutto ciò che richiede o che il team ne abbia un'esperienza pregressa. Così facendo è possibile anticipare eventuali rilavorazioni dovute a disallineamenti tra ciò che il cliente dice di volere e ciò che il cliente effettivamente vuole e chiare i dubbi prima dell'inizio della progettazione.

⁵⁴ Per approfondimenti vedasi paragrafo 1.3.4.

Per rendere evidenti questa tipologia di situazioni è stato introdotto il seguente standard [Figura 36], che consente di identificare in modo istantaneo le diverse categorie di requisiti:

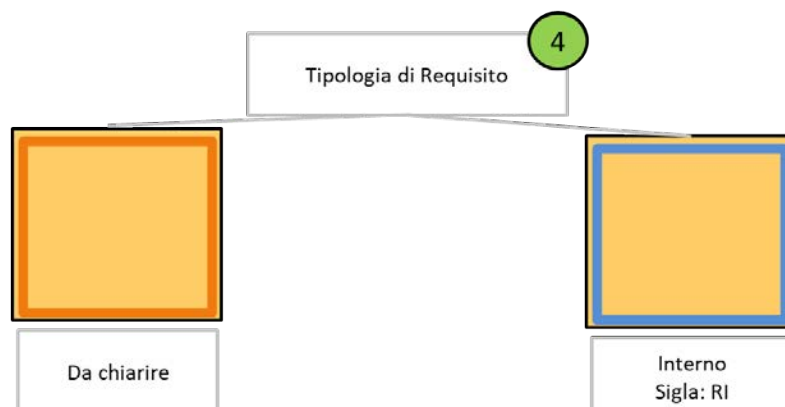


Figura 36 - Standard tipologie di requisiti - Fonte: auxiell

La formalizzazione dei requisiti in uno strumento visual consente di identificare fin da subito quelli che sono i potenziali “gruppi critici” per la macchina che si andrà a progettare, ovvero parti della macchina per le quali il cliente non è in grado di dare una chiara definizione dei propri bisogni o per le quali il team non possiede esperienza pregressa formalizzata nel know-how aziendale sotto forma di mattoni tecnologici.

L’identificazione anticipata dei potenziali gruppi critici consente al team di assegnare loro una priorità elevata, arrivando in breve tempo alla progettazione e al montaggio al fine di riprodurre le principali funzionalità identificate e in seguito di testarlo con il cliente. Questo consente sia al team che al cliente stesso di acquisire conoscenza, in modo da evitare rilavorazioni in fase avanzata di progetto.

Una volta che tutti i requisiti sono stati formalizzati in cartellini post-it, il team procede alla prioritizzazione di essi all'interno del tabellone dei requisiti. Il tabellone dei requisiti assume la struttura di una matrice di priorità-sforzo e lo standard che il team segue per la loro categorizzazione è il seguente [Figura 37]:

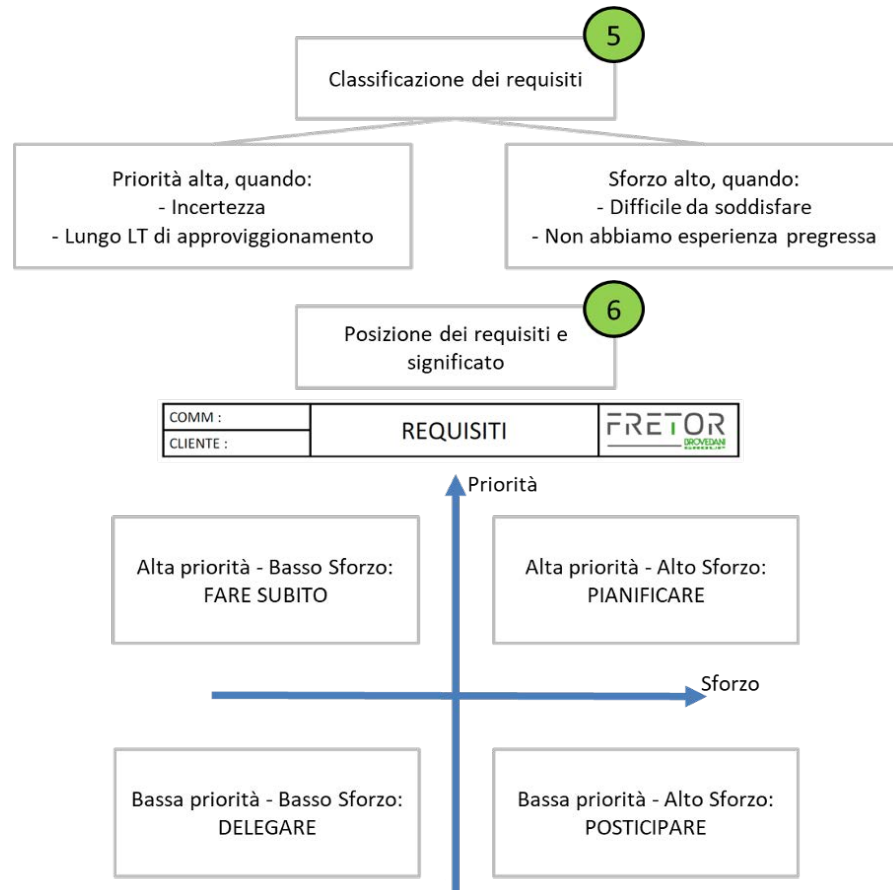


Figura 37 - Standard di classificazione e posizionamento dei requisiti - Fonte: auxiell

Quando i requisiti sono stati posizionati all'interno del tabellone dei requisiti (esempio riportato in [Figura 39]), il team dovrà procedere con:

- Acquisire conoscenza relativa ai requisiti da chiarire, attraverso il coinvolgimento del cliente;
- Esploredere i requisiti in storie, le quali rappresentano le funzionalità del prodotto espresse da un punto di vista tecnico, ovvero con il linguaggio del team di sviluppo;

Lo standard di compilazione delle storie all'interno del Backlog è riportato in seguito [Figura 38].

La completezza del Backlog è in parte dipendente dalla complessità della macchina. Nel caso di progetti tecnologicamente complessi il team procederà con l'esplosione in storie di requisiti che fanno riferimento a moduli della macchina caratterizzati da priorità più elevata, per poi procedere in fasi successive alla definizione delle rimanenti storie. Nel caso di commesse

tecnologicamente meno complesse, il team procede esplodendo tutte le storie nel Backlog, comprese quelle relative al montaggio dei gruppi.

A questo punto il team ha dato vita al Product Backlog, che è uno dei requisiti fondamentali per far partire la pianificazione del progetto utilizzando il framework Scrum. Un esempio di applicazione del framework a partire da un Backlog già formalizzato è oggetto del successivo paragrafo.

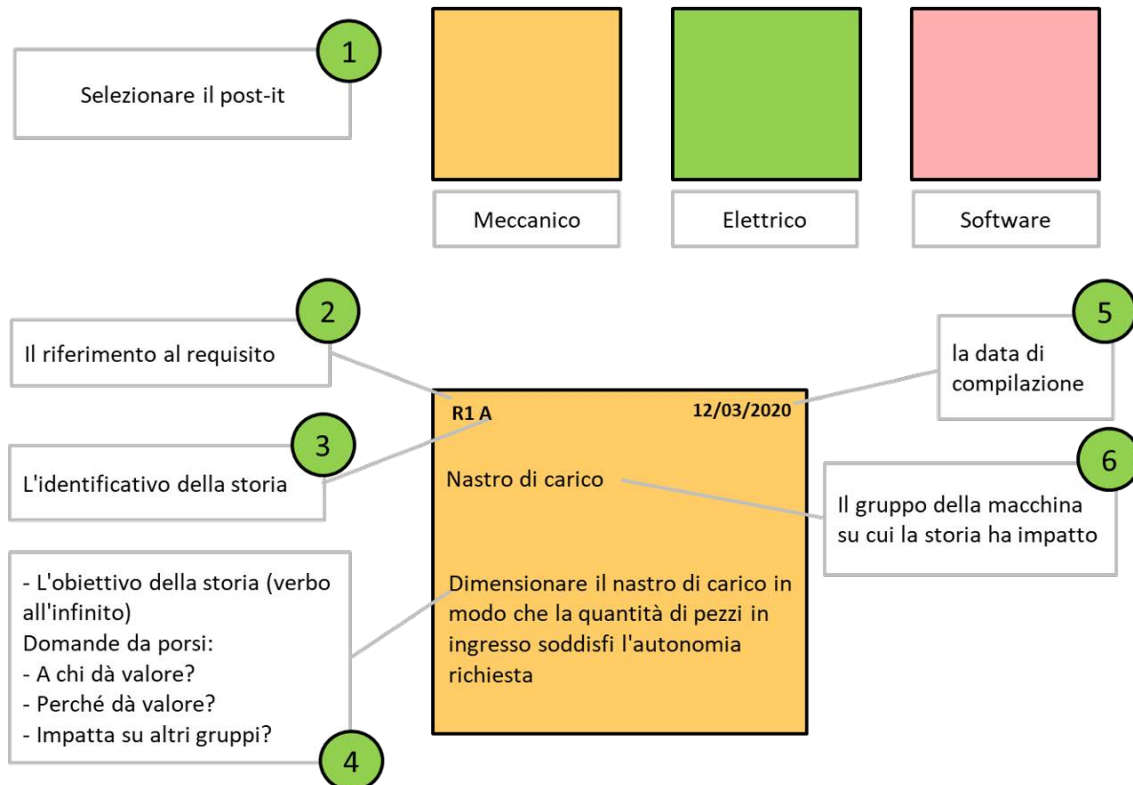


Figura 38 - Standard di creazione delle storie - Fonte: auxiell

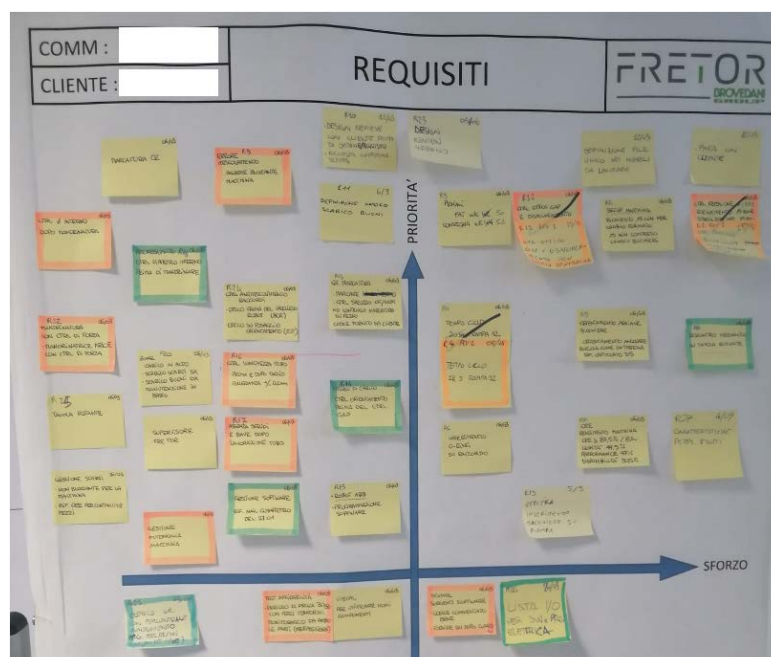


Figura 39 - Esempio tabellone dei requisiti - Fonte: FreTor

4.3 Esempio di svolgimento di uno Sprint in FreTor

Al momento della stesura dell'elaborato in FreTor ci sono tre progetti di sviluppo attivi gestiti attraverso la pianificazione basata su Scrum. Di seguito è riportato lo svolgimento di uno Sprint relativo ad una di queste commesse.

Per tutti i progetti gestiti con il framework, la durata dello Sprint è sincronizzata ad una settimana lavorativa.

Tipicamente le riunioni di Sprint Review e di Sprint Planning vengono tutte svolte il venerdì nel primo pomeriggio, mentre gli eventi di Daily Scrum sono svolti tutte le mattine a partire dalle otto.

Nella prima parte del progetto di trasformazione lean gli eventi previsti dal framework erano supportati da un consulente, mentre in una seconda fase il ruolo dello Scrum Master è passato a due figure interne che alternandosi soppressiedono alle varie sedute. In assenza di tali figure sono gli altri membri del team che a turno “tirano” i vari eventi e si impegnano a far rispettare gli standard formalizzati; in questo modo è possibile sensibilizzare tutti i membri del team ad acquisire conoscenza ed esperienza riguardo alle logiche del framework e agli standard previsti nei diversi eventi.

Sprint Planning.

Al termine della riunione di Sprint Review dello Sprint precedente, il team dà avvio al meeting di pianificazione del nuovo Sprint. Ciascun membro del team preleva dal Product Backlog (evidenziato in [Figura 40]) le storie che decide di pianificare per lo Sprint in corso in funzione dei valori di priorità-sforzo e di eventuali vincoli di precedenza tra i diversi deliverable.

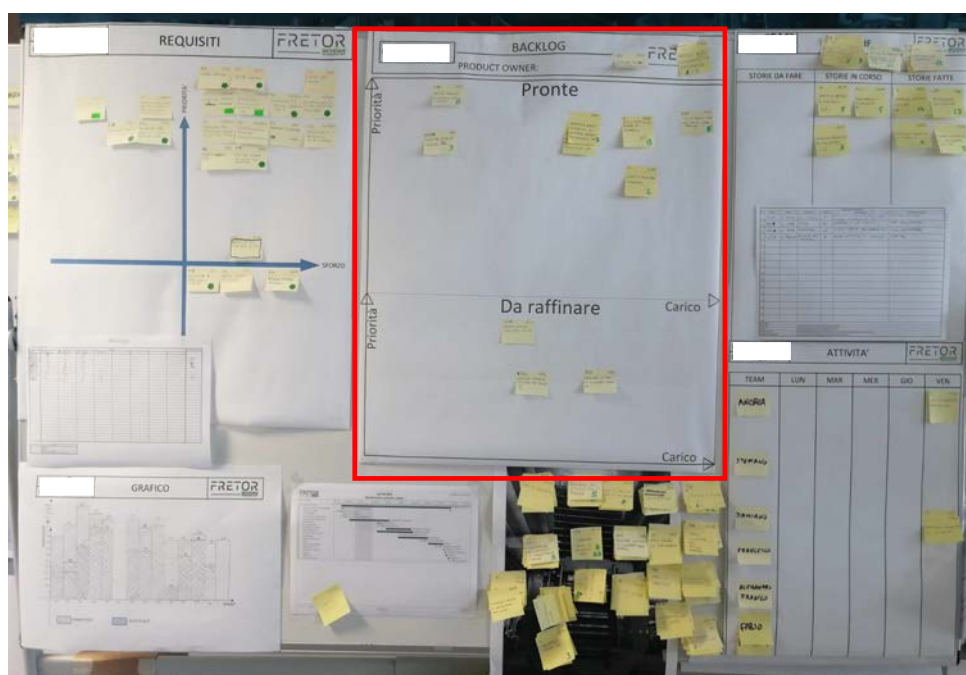


Figura 40 - Tabelloni Visual adottati in FreTor in cui è evidenziato il Product Backlog - Fonte: FreTor

In questa riunione di Sprint Planning il team si fa carico di completare sette storie, di cui alcune vengono riportate in seguito a titolo esemplificativo [Figura 41]:

R1	08/09/2020		08/09/2020	R8A	28/08/2020
VERIFICA MANUALE DI MONTAGGIO 2/2		DISEGNO PIASTRA CON SPESSORAMENTO MACCHINA		PROGETTAZIONE GR0531319	
5		5		13	

Figura 41 - Esempio Storie selezionate in FreTor - Fonte: FreTor

Il totale dei punti target pianificati per lo Sprint in corso è di quarantasette punti.

Dopo aver inserito le storie all'interno del Tabellone dello Sprint, ciascun membro del team si focalizza sul pianificare i task per il primo giorno di Sprint all'interno del Tabellone dei Task.

Lo standard di creazione dei Task prevede che ciascuno di essi debba essere completato all'interno della giornata in cui è pianificato, altrimenti è necessario suddividerlo in più task.

Daily Scrum.

Ogni giorno dello Sprint il team si riunisce di fronte al tabellone relativo alla commessa per un meeting della durata massima di dieci minuti.

All'inizio del meeting, lo Scrum Master o un altro membro del team riportano le presenze e i minuti di ritardo in un apposito modulo [Figura 42]:

DATA	STEFANO	MATEO B	OTAZ	ALESSANDRO	ROBERTO	FRANCESCO	MARCO C.	DURATA SCRUM
06/09	✓	✓	✓	✓	✓			20'
07/09	✓	+7	✓	✓	✓			8'
08/09	✓	✓	✓	✓	✓			5'
09/09	✓	✓	✓	✓	✓	✓		15'
10/09	✓	✓	✓	✓	✓	✓	✓	20'
11/09	✓	✓	✓	✓	✓	✓	✓	15'
12/09	✓	✓	✓	✓	✓	✓	✓	15'
13/09	✓	✓	✓	✓	✓	✓	✓	15'
14/09	✓	✓	✓	✓	✓	✓	✓	15'
15/09	✓	✓	✓	✓	✓	✓	✓	15'
16/09	✓	✓	✓	✓	✓	✓	✓	15'
17/09	✓	✓	✓	✓	✓	✓	✓	15'
18/09	✓	✓	✓	✓	✓	✓	✓	15'
19/09	✓	✓	✓	✓	✓	✓	✓	15'
20/09	✓	✓	✓	✓	✓	✓	✓	15'
21/09	✓	✓	✓	✓	✓	✓	✓	15'
22/09	✓	✓	✓	✓	✓	✓	✓	15'
23/09	✓	✓	✓	✓	✓	✓	✓	15'
24/09	✓	✓	✓	✓	✓	✓	✓	15'
25/09	✓	✓	✓	✓	✓	✓	✓	15'
26/09	✓	✓	✓	✓	✓	✓	✓	15'
27/09	✓	✓	✓	✓	✓	✓	✓	15'
28/09	✓	✓	✓	✓	✓	✓	✓	15'
29/09	✓	✓	✓	✓	✓	✓	✓	15'
30/09	✓	✓	✓	✓	✓	✓	✓	15'
01/10	✓	✓	✓	✓	✓	✓	✓	15'
02/10	✓	✓	✓	✓	✓	✓	✓	15'
03/10	✓	✓	✓	✓	✓	✓	✓	15'
04/10	✓	✓	✓	✓	✓	✓	✓	15'
05/10	✓	✓	✓	✓	✓	✓	✓	15'
06/10	✓	✓	✓	✓	✓	✓	✓	15'
07/10	✓	✓	✓	✓	✓	✓	✓	15'
08/10	✓	✓	✓	✓	✓	✓	✓	15'
09/10	✓	✓	✓	✓	✓	✓	✓	15'
10/10	✓	✓	✓	✓	✓	✓	✓	15'
11/10	✓	✓	✓	✓	✓	✓	✓	15'
12/10	✓	✓	✓	✓	✓	✓	✓	15'
13/10	✓	✓	✓	✓	✓	✓	✓	15'
14/10	✓	✓	✓	✓	✓	✓	✓	15'
15/10	✓	✓	✓	✓	✓	✓	✓	15'
16/10	✓	✓	✓	✓	✓	✓	✓	15'
17/10	✓	✓	✓	✓	✓	✓	✓	15'
18/10	✓	✓	✓	✓	✓	✓	✓	15'
19/10	✓	✓	✓	✓	✓	✓	✓	15'
20/10	✓	✓	✓	✓	✓	✓	✓	15'
21/10	✓	✓	✓	✓	✓	✓	✓	15'
22/10	✓	✓	✓	✓	✓	✓	✓	15'
23/10	✓	✓	✓	✓	✓	✓	✓	15'
24/10	✓	✓	✓	✓	✓	✓	✓	15'
25/10	✓	✓	✓	✓	✓	✓	✓	15'
26/10	✓	✓	✓	✓	✓	✓	✓	15'
27/10	✓	✓	✓	✓	✓	✓	✓	15'
28/10	✓	✓	✓	✓	✓	✓	✓	15'
29/10	✓	✓	✓	✓	✓	✓	✓	15'
30/10	✓	✓	✓	✓	✓	✓	✓	15'
31/10	✓	✓	✓	✓	✓	✓	✓	15'
01/11	✓	✓	✓	✓	✓	✓	✓	15'
02/11	✓	✓	✓	✓	✓	✓	✓	15'
03/11	✓	✓	✓	✓	✓	✓	✓	15'
04/11	✓	✓	✓	✓	✓	✓	✓	15'
05/11	✓	✓	✓	✓	✓	✓	✓	15'
06/11	✓	✓	✓	✓	✓	✓	✓	15'
07/11	✓	✓	✓	✓	✓	✓	✓	15'
08/11	✓	✓	✓	✓	✓	✓	✓	15'
09/11	✓	✓	✓	✓	✓	✓	✓	15'
10/11	✓	✓	✓	✓	✓	✓	✓	15'
11/11	✓	✓	✓	✓	✓	✓	✓	15'
12/11	✓	✓	✓	✓	✓	✓	✓	15'
13/11	✓	✓	✓	✓	✓	✓	✓	15'
14/11	✓	✓	✓	✓	✓	✓	✓	15'
15/11	✓	✓	✓	✓	✓	✓	✓	15'
16/11	✓	✓	✓	✓	✓	✓	✓	15'
17/11	✓	✓	✓	✓	✓	✓	✓	15'
18/11	✓	✓	✓	✓	✓	✓	✓	15'
19/11	✓	✓	✓	✓	✓	✓	✓	15'
20/11	✓	✓	✓	✓	✓	✓	✓	15'
21/11	✓	✓	✓	✓	✓	✓	✓	15'
22/11	✓	✓	✓	✓	✓	✓	✓	15'
23/11	✓	✓	✓	✓	✓	✓	✓	15'
24/11	✓	✓	✓	✓	✓	✓	✓	15'
25/11	✓	✓	✓	✓	✓	✓	✓	15'
26/11	✓	✓	✓	✓	✓	✓	✓	15'
27/11	✓	✓	✓	✓	✓	✓	✓	15'
28/11	✓	✓	✓	✓	✓	✓	✓	15'
29/11	✓	✓	✓	✓	✓	✓	✓	15'
30/11	✓	✓	✓	✓	✓	✓	✓	15'
01/12	✓	✓	✓	✓	✓	✓	✓	15'
02/12	✓	✓	✓	✓	✓	✓	✓	15'
03/12	✓	✓	✓	✓	✓	✓	✓	15'
04/12	✓	✓	✓	✓	✓	✓	✓	15'
05/12	✓	✓	✓	✓	✓	✓	✓	15'
06/12	✓	✓	✓	✓	✓	✓	✓	15'
07/12	✓	✓	✓	✓	✓	✓	✓	15'
08/12	✓	✓	✓	✓	✓	✓	✓	15'
09/12	✓	✓	✓	✓	✓	✓	✓	15'
10/12	✓	✓	✓	✓	✓	✓	✓	15'
11/12	✓	✓	✓	✓	✓	✓	✓	15'
12/12	✓	✓	✓	✓	✓	✓	✓	15'
13/12	✓	✓	✓	✓	✓	✓	✓	15'
14/12	✓	✓	✓	✓	✓	✓	✓	15'
15/12	✓	✓	✓	✓	✓	✓	✓	15'
16/12	✓	✓	✓	✓	✓	✓	✓	15'
17/12	✓	✓	✓	✓	✓	✓	✓	15'
18/12	✓	✓	✓	✓	✓	✓	✓	15'
19/12	✓	✓	✓	✓	✓	✓	✓	15'
20/12	✓	✓	✓	✓	✓	✓	✓	15'
21/12	✓	✓	✓	✓	✓	✓	✓	15'
22/12	✓	✓	✓	✓	✓	✓	✓	15'
23/12	✓	✓	✓	✓	✓	✓	✓	15'
24/12	✓	✓	✓	✓	✓	✓	✓	15'
25/12	✓	✓	✓	✓	✓	✓	✓	15'
26/12	✓	✓	✓	✓	✓	✓	✓	15'
27/12	✓	✓	✓	✓	✓	✓	✓	15'
28/12	✓	✓	✓	✓	✓	✓	✓	15'
29/12	✓	✓	✓	✓	✓	✓	✓	15'
30/12	✓	✓	✓	✓	✓	✓	✓	15'
31/12	✓	✓	✓	✓	✓	✓	✓	15'

Figura 42 - Modulo di reportistica del Daily Scrum - Fonte: FreTor

Successivamente, ciascun membro del Team ispeziona il Tabellone dello Sprint spostando il post-it relativo alla storia di cui è owner nella colonna “Storie in corso”, nel caso in cui siano iniziate le attività relative a quel deliverable, o nella colonna “Storie fatte”, nel caso in cui il deliverable definito dalla storia sia stato raggiunto.

In seguito, ad uno ad uno i membri del team ispezionano il Tabellone dei Task: nel caso in cui il task pianificato per il giorno precedente sia stato completato, allora viene rimosso dal tabellone. Nel caso in cui il task non sia stato completato, l'owner del task scrive una segnalazione all'interno della *tabella delle traslazioni* [Figura 43] e successivamente il team pianifica che contromisure adottare per recuperare il ritardo e cercare di impedire che la relativa storia risulti non completata.

N°	DATA	NOME	ATTIVITA'	TEMPO	TRASLAZIONE ATTIVITA'	COME RECUPERARE
					DESCRIZIONE	
1	28/08		SCI. PNEUM.	8	ATTIVITA' DI CODIFICA PARTICOLARI SU COMPTON E LAUREO COMPTON	NON RECUPERATO
2	09/09		CULLE	12	CONFRONTO CON CLIENTE	NON RECUPERATO
3	09/09		PROTEZIONI	12	FERIE E RITARDI PROGRAMMATIONE	NON RECUPERATO
4	11/09		DEFINIRE MOD. COLLAUDO	8	ALTRE ATTIVITA' DI COMPTON	NON REC.
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						

Figura 43 - Tabella delle traslazioni per la commessa in analisi- Fonte: FreTor

La tabella delle traslazioni è uno strumento voluto dal Team Scrum per poter tenere traccia delle problematiche che hanno generato degli slittamenti nei task e le contromisure adottate.

Durante i Daily Scrum è stato riscontrato che uno dei principali impedimenti nel portare a termine i task è relativo al fatto che una risorsa coinvolta nel progetto non è completamente dedicata ad esso e che spesso si generano delle situazioni di urgenza verso altre commesse che si trovano in fasi più avanzate dello sviluppo.

Dopo aver ispezionato le variazioni rispetto al pianificato e aver individuato eventuali contromisure, ciascun membro del team pianifica le attività per la giornata successiva dello Sprint.

Al termine dell'evento di Daily Scrum, lo Scrum Master o un altro membro del team riportano la durata del meeting nel modulo riportato in [Figura 42].

Sprint Review.

A fine Sprint, il team Scrum si riunisce di fronte al tabellone della commessa per la riunione di Sprint Review. Non tutti i membri del team partecipano in presenza, in quanto alcuni sono in collegamento tramite videochiamata perché si trovano in trasferta presso un'altra sede aziendale.

Di seguito è riportato il dettaglio del Tabellone dello Sprint all'inizio della riunione [Figura 44]:



Figura 44 - Tabellone dello Sprint ad inizio Sprint Review - Fonte: FreTor

Per ciascuna delle storie presenti all'interno della sezione "Storie Fatte" i membri del team condividono brevemente quello che è stato fatto e in seguito il Project Manager valida il completamento di tali storie. Una delle storie pianificate è rimasta nella colonna "Storie in Corso" durante lo Sprint concluso e perciò è spostata nella colonna delle "Storie da Fare" per il successivo Sprint.

Il punteggio totale accumulato dal team nel corso dello Sprint è di quarantadue punti su quarantasette pianificati, i quali vengono riportati nel grafico di velocità degli Sprint [Figura 45]:

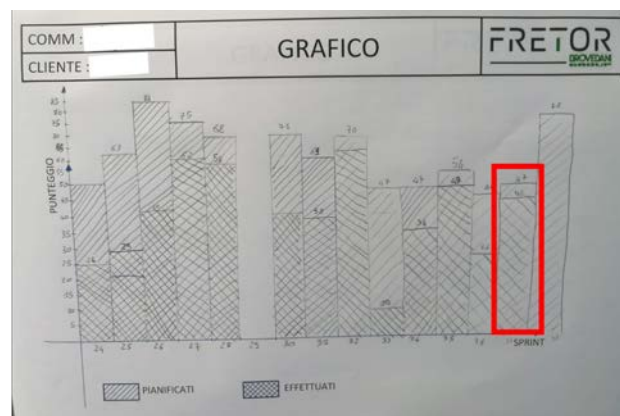


Figura 45 - Grafico di velocità degli Sprint - Fonte: FreTor

Nella parte finale della riunione di Sprint Review il team è coinvolto in un breve brainstorming sulle principali cause che provocano deviazioni rispetto a quanto pianificato. Di seguito è riportato quanto emerso:

In primo luogo, il team ha espresso la propria difficoltà nel pianificare le attività quando non è raro che vi siano delle situazioni di maggior priorità in altre commesse che impediscono di dedicarsi completamente allo svolgimento delle attività pianificate.

In secondo luogo, il team ha manifestato delle difficoltà nel quantificare il carico di lavoro delle attività, soprattutto nel caso in cui si tratti di qualcosa di nuovo o di non prevedibile. Questo secondo aspetto ha portato ad una riflessione critica tra Scrum Master e gli altri membri del team in riferimento alla metodologia con la quale viene assegnato il punteggio alla storia, evidenziando un disallineamento tra lo standard condiviso e quanto fatto dal team. La tendenza è quella di assegnare il punteggio della storia in funzione di quello che è il carico di lavoro previsto in termini di ore o di risorse coinvolte per portarla a termine e non confrontando le varie storie tra di loro.

Un ulteriore aspetto riguarda il coinvolgimento dei fornitori di servizi o di materiali, i quali in alcuni casi provocano uno slittamento indesiderato delle attività pianificate e in alcuni casi il non completamento di alcune storie.

Infine, è emerso che l'operazione di ispezione del Backlog risulta difficoltosa per la presenza di un numero elevato di storie e per il fatto che vi sia difficoltà ad individuare le tre macro-categorie di deliverable: meccanici, elettrici e software. In modo da evitare confusione nella ricerca delle storie si è deciso di modificare lo standard precedentemente adottato e di introdurre diversi colori nei post-it per identificare meglio la categoria progettuale a cui appartiene la storia (il nuovo standard di creazione delle storie è riportato in [Figura 40]).

Alcune delle problematiche sopra descritte sono state riportate come segnalazione nel tabellone Problem Solving comune a tutti i team di progettazione in FreTor, in modo che la causa radice possa essere individuata ed eliminata.

Sprint Planning – Sprint successivo.

Dopo aver concluso la riunione di Review con la sessione di brainstorming, il team ha dato avvio al nuovo Sprint con l'evento di Sprint Planning. Ciascuno dei membri del team ha selezionato alcune delle voci del Backlog portandole all'interno del Tabellone dello Sprint. In questa particolare sessione di Sprint Planning è stata introdotta nel team anche la figura del responsabile del montaggio, in quanto la commessa si trova in una fase avanzata di progettazione e il Project Manager ha espresso la necessità di iniziare con il montaggio di alcuni gruppi della macchina. Inoltre, è stata condivisa tra i

membri del team la necessità di innalzare la velocità in modo da potersi allineare con la successiva milestone di progetto riportata nel Gantt.

Dopo aver pianificato le storie, il team ha proseguito pianificando i task all'interno del Tabellone dei task per il primo giorno dello Sprint (non vi è documentazione fotografica a supporto).

L'aspetto che assume il Tabellone dello Sprint in seguito alla riunione di pianificazione è il medesimo [Figura 46]:



Figura 46 - Tabellone dello Sprint al termine della seconda riunione di Sprint Planning - Fonte: FreTor

In totale i punti pianificati dal team per lo Sprint che è iniziato sono settantotto e sono stati riportati all'interno del grafico di velocità degli Sprint [Figura 47]:

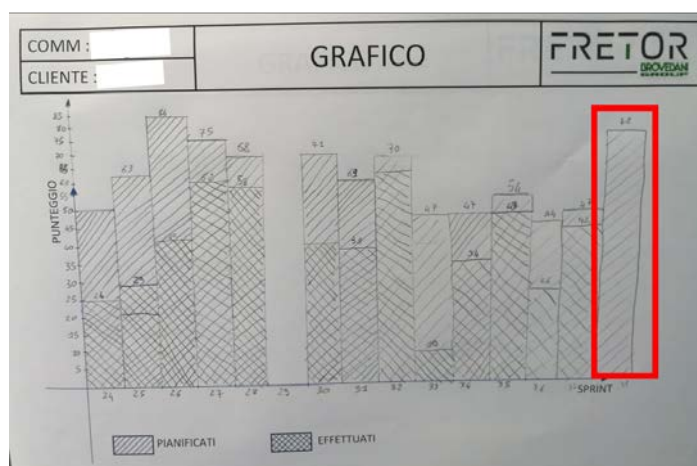


Figura 47 - Grafico di velocità dello Sprint aggiornato al secondo Sprint descritto - Fonte: FreTor

Una delle criticità emerse durante la seduta di Sprint Planning riguarda lo standard di creazione delle storie. Lo standard prevede che la storia debba essere completata entro la durata dello Sprint e nel caso in cui non sia possibile farlo essa debba essere spezzata in più storie.

La soluzione adottata dal team prevede la creazione di più storie con lo stesso nome e un numero che ne indichi la partizione, come esemplificato in seguito [Figura 48]:



Figura 48 - Esempio di suddivisione delle storie

Da un lato questa pratica permette di poter concludere la storia nell'arco della durata dello Sprint, coerentemente con quanto definito nello standard, ma allo stesso tempo il deliverable non è testabile, in quanto non individuabile in modo oggettivo. Lo Scrum Master ha condiviso con il team la necessità di rispettare lo standard per poter praticare i principi di ispezione e adattamento riguardo al contenuto della storia, indicando anche all'interno della storia "parziale" (Verifica manuale di montaggio $\frac{1}{2}$) quello che può essere un deliverable che ne certifichi il completamento.

L'analisi del caso studio appena discusso è riportata all'interno del capitolo seguente dell'elaborato.

Capitolo 5.

Modello di implementazione, Analisi critica e Conclusioni

“Without a goal you can’t score”

Casey Neistat

Nel seguente capitolo è riportata l’analisi del processo adottato nei casi studio esposti nei due capitoli precedenti dell’elaborato, a partire dai quali sono stati formulati un modello che descrive la struttura di un processo per l’implementazione dell’approccio Hybrid Stage - Gate e un diagramma di flusso che formalizza le attività per lo svolgimento degli eventi di Scrum.

Successivamente sono riportate l’analisi e alcune considerazioni delle difficoltà incontrate nell’implementazione del framework e gli adattamenti riscontrati nell’implementazione all’interno dei casi studio.

Infine, sono riportate conclusioni e considerazioni personali sul lavoro svolto nel presente elaborato.

5.1 Analisi dei casi studio

L'approccio adottato nei due casi studio descritti all'interno dell'elaborato segue le logiche del modello Hybrid Stage - Gate esposto nel secondo capitolo. Di seguito sono illustrate delle considerazioni riguardo l'organizzazione e la progettazione dei processi in ottica Hybrid Stage-Gate, generate a partire dall'analisi dei due casi studio.

Come esposto nell'analisi di letteratura del secondo capitolo, il modello Stage - Gate fornisce all'intero processo una struttura solida, necessaria per portare a termine progetti complessi quali lo sviluppo di prodotti tangibili, garantendo supporto per la pianificazione di lungo termine e il monitoraggio delle milestones di progetto a livello strategico.

Il numero di Stage è variabile in funzione della rigidità che si vuole conferire al processo:

- Nel primo caso studio il numero di Stage è cinque, che corrisponde alla definizione della specifica di prodotto, sviluppo preliminare, sviluppo di dettaglio, alfa test, industrializzazione e produzione.
- Nel secondo caso studio gli Stage si riducono a tre, caratterizzati dalle fasi di definizione della specifica di prodotto ed elaborazione dell'offerta, progettazione e infine approvvigionamento e produzione.

I due casi studio sono rappresentativi di due modelli di gestione dei processi differenti: nel primo caso il processo di sviluppo è separato e antecedente rispetto al processo logistico - produttivo, tipico di un modello di gestione MTS/MTO. Nel secondo caso l'azienda adotta un approccio ETO, prevedendo la progettazione e la produzione come fasi del progetto.

Dall'analisi dei casi studio è quindi possibile dedurre che il modello Hybrid Stage-Gate è applicabile a processi che fanno riferimento a modelli organizzativi differenti.

I due casi studio esposti presentano inoltre delle comunanze a livello di progettazione del processo, le quali sono analizzate di seguito.

Il primo Stage del processo è orientato alla definizione della specifica di prodotto, la quale racchiude le informazioni fino a quel momento conosciute riguardo al prodotto da sviluppare. La specifica di prodotto è formalizzata in un documento e non è congelata, ma può evolvere in funzione della nuova conoscenza che viene acquisita nel corso del progetto. Risulta fondamentale quindi la progettazione di uno standard di analisi e aggiornamento della stessa. Nel caso del modello di gestione ETO, la specifica di prodotto è generata a

partire da una richiesta da parte del cliente, mentre nel caso di un modello di gestione MTS/MTO risultano necessarie attività di acquisizione della VoC⁵⁵.

Per quanto riguarda il primo Gate del processo, in entrambi i casi studio esso è strutturato con una modalità Go/Kill ed è focalizzato su:

- Approvazione dell'offerta comprensiva della specifica di prodotto da parte del cliente, nel caso di processo basato su modello di gestione ETO;
- Approvazione della specifica di prodotto e inserimento all'interno della Product RoadMap da parte del team strategico, nel caso di processo basato su modello di gestione MTO/MTS.

Il superamento del primo Gate sancisce ufficialmente il kick-off del progetto.

Una volta che la direzione/team strategico dà avvio al progetto, segue l'attività di investitura del Product Owner e formazione del Development Team. La composizione del team di sviluppo è funzione delle competenze fino a quel momento ritenute necessarie per portare a termine l'intero progetto.

Le mansioni di ciascun membro del team e lo svolgimento degli eventi di Scrum descritti all'interno dei casi studio sono stati formalizzati in un diagramma di flusso [Figura 49-50-51], che in seguito viene esposto.

Una volta formato il team, esso può occuparsi di dar vita al Product Backlog e dar avvio alla gestione del progetto tramite l'utilizzo del framework Scrum.

L'attività di creazione del Product Backlog a partire dai requisiti formalizzati nella specifica di prodotto è uno degli input al processo di pianificazione dei progetti con Scrum ed è rappresentata dal ramo di sinistra (denominato "Esame e classificazione dei requisiti di prodotto/processo") in ingresso al diagramma di flusso [Figura 49].

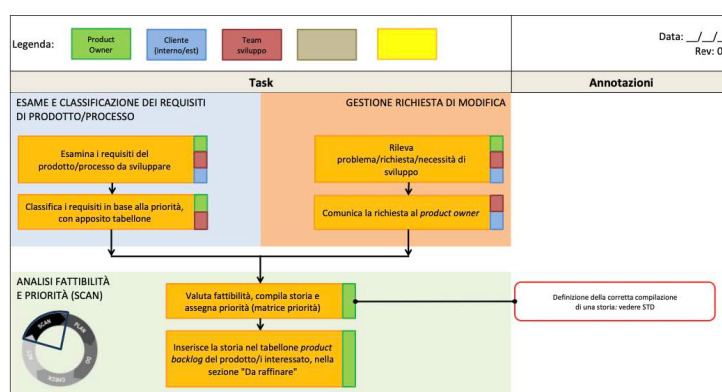


Figura 49 - Flowchart utilizzo framework Scrum - Fonte: auxiell

⁵⁵ *Voice of the Customer*: "Il VOC è un termine usato per descrivere il processo di acquisizione dei dati inerenti le aspettative, le preferenze e le obiezioni dei clienti. In breve, è un concetto che racchiude tutto ciò che il cliente desidera o di cui ha bisogno, i benefici e le aspettative associati ad un prodotto o ad un servizio, il livello di soddisfazione del cliente stesso." Fonte: <https://www.salesclub.it/news/il-potere-di-voc-l-ascolto-efficace-della-voce-del-cliente.html>

La definizione della specifica di prodotto diventa input al processo di gestione con Scrum in quanto è il raccoglitore di quelli che sono i requisiti del prodotto/processo da sviluppare; la specifica viene elaborata dal Product Owner e dal team di sviluppo al fine di inserire le storie all'interno del Backlog di prodotto. È possibile utilizzare lo strumento del tabellone dei requisiti (come illustrato nel secondo caso studio) o utilizzare direttamente il Backlog (come nel primo caso studio).

L'input al flowchart che si origina dal ramo di destra (denominato "Gestione richiesta modifica") rappresenta le situazioni in cui l'acquisizione di nuova conoscenza nel corso del progetto (che può essere generata da un problema, richiesta di modifica o necessità di sviluppo) genera una variazione al contenuto del Product Backlog.

Una volta che a partire dalla specifica di prodotto il team ha popolato il Product Backlog, è possibile dare avvio formalmente alla pianificazione del progetto a livello micro attraverso l'approccio iterativo definito dal framework Scrum, descritto in [Figura 50-51].

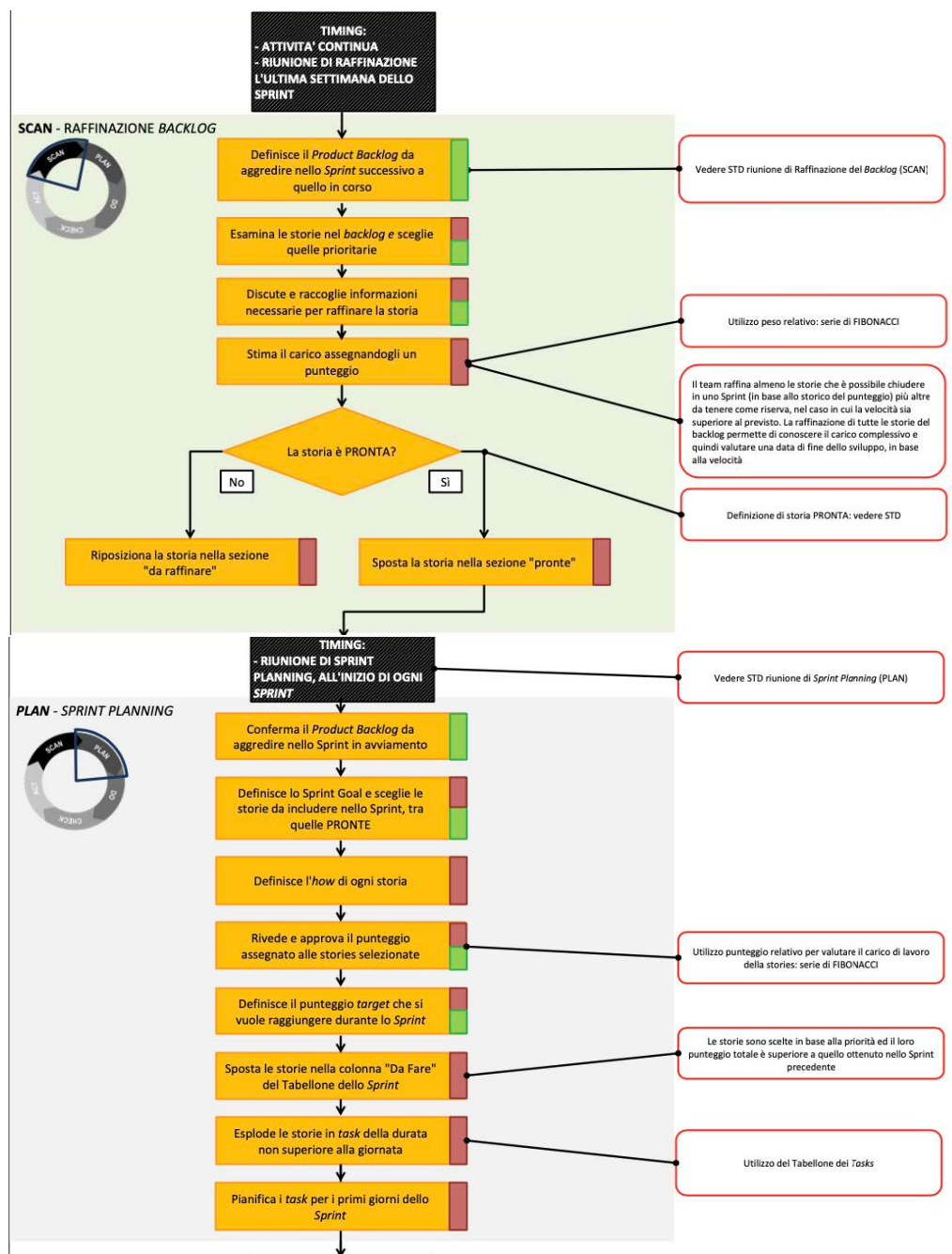


Figura 50 - Flowchart utilizzo framework Scrum - Fonte: auxiell

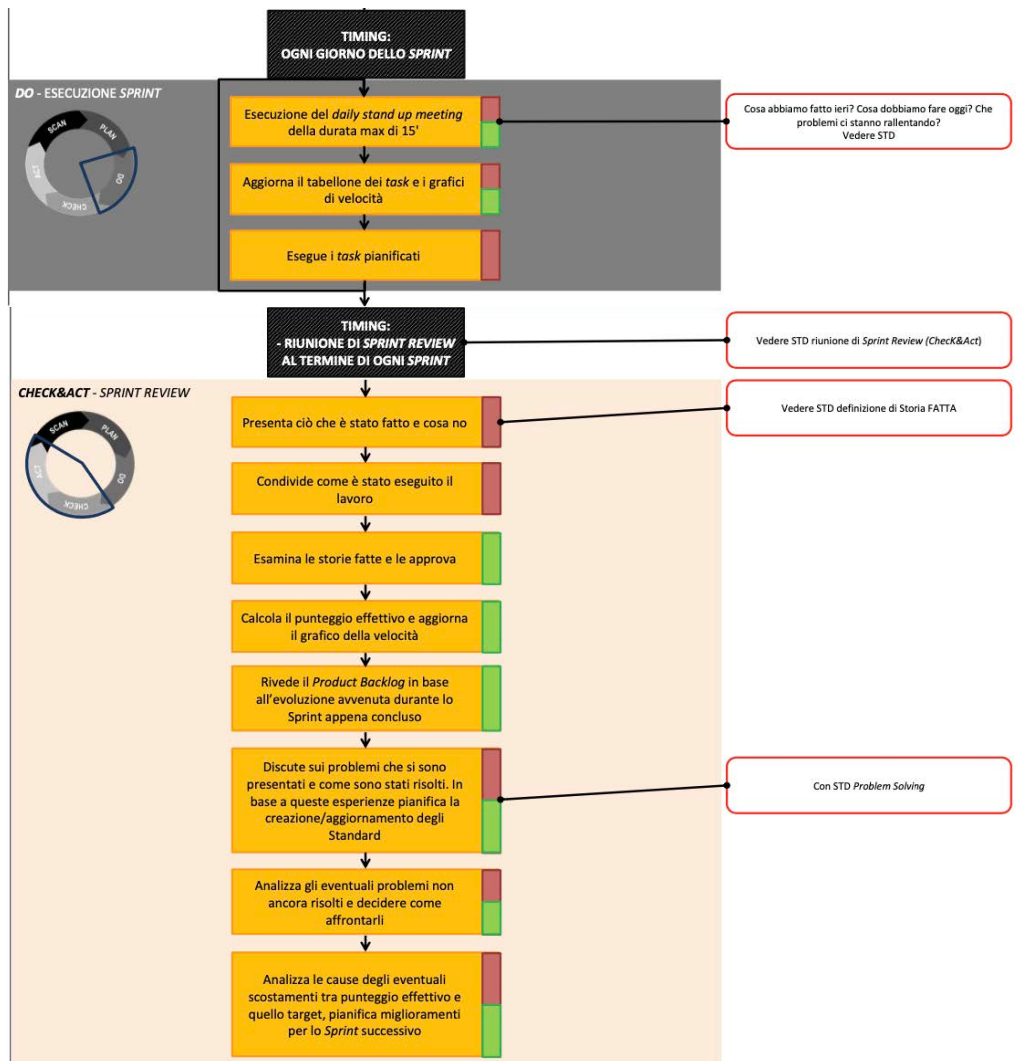


Figura 51 - Flowchart utilizzo framework Scrum - Fonte: auxiell

5.2 Modello di implementazione

L'analisi dei casi studio esposta nel paragrafo precedente dell'elaborato ha permesso di formulare quella che può rappresentare una struttura comune a tutti i processi per favorire l'implementazione del modello Hybrid Stage - Gate, la quale viene rappresentata graficamente come segue [Figura 52].

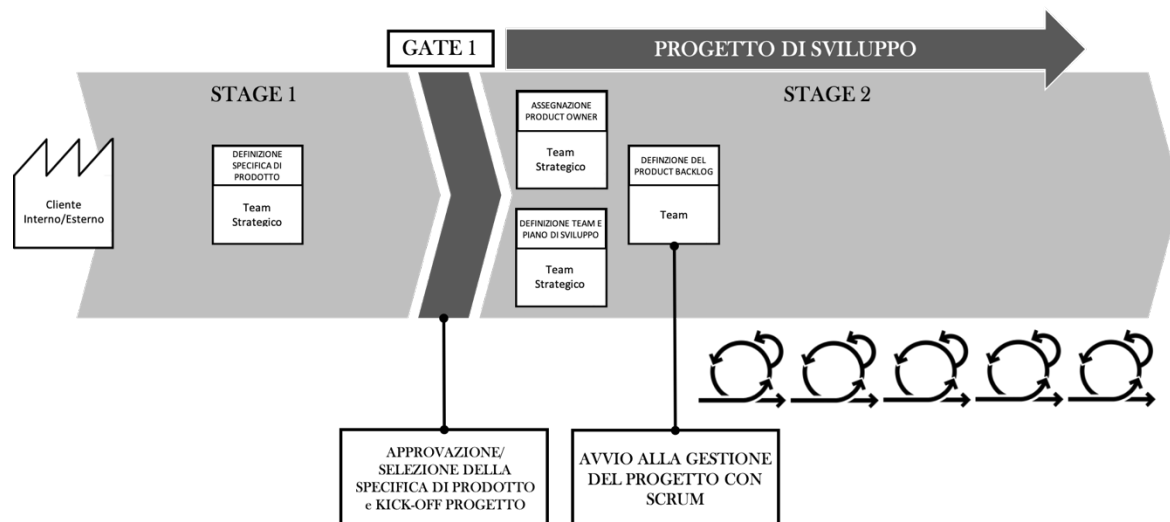


Figura 52 - Modello di processo proposto per l'implementazione di Hybrid Stage-Gate

Il modello riportato in [Figura 52] si pone l'obiettivo di identificare l'ossatura minima che il processo di development deve prevedere per poter rendere operativa l'implementazione del modello Hybrid Stage-Gate.

All'interno del primo Stage non sono state riportate altre attività oltre alla creazione della specifica di prodotto perché, come visto nei due casi studio, esse possono variare in funzione della specifica realtà aziendale.

Ad esempio, nel secondo caso studio già all'interno del primo Stage viene valutata la fattibilità del prodotto per poi procedere alla generazione dell'offerta, mentre nel primo caso studio l'analisi di fattibilità avviene all'interno del secondo Stage, ma all'interno della prima fase sono presenti attività quali l'acquisizione dei bisogni dei clienti e la creazione della Product Roadmap.

Nel modello è rappresentato solamente il primo Gate, il quale ha la funzione di formalizzare l'avvio del progetto di sviluppo. Non sono state riportate altre attività in quanto, come per quanto descritto per il primo stage, la struttura del primo Gate (e più in generale di tutti i Gate) dovrà essere definita in funzione degli specifici bisogni aziendali.

Ad esempio, il processo di sviluppo di un'azienda all'interno del contesto automotive/aeronautico dovrà prevedere dei controlli rigidi ai Gate per garantire i requisiti minimi di qualità richiesti dagli appositi enti.

All'interno di un modello Hybrid Stage-Gate il team non dovrà però adottare un approccio passivo nei confronti dei Gate, ma considerare già all'interno del Backlog di prodotto i deliverable necessari al superamento di essi. In questo modo è possibile superare l'approccio statico e rigido dettato dalla visione tradizionale dei Gate e adottare un approccio più dinamico che si fonde con il ciclo di miglioramento SPDCA, considerando il Gate come una fase di check e non come un ostacolo.

Per quanto riguarda il secondo Stage del modello, sono inserite unicamente le attività preliminari necessarie a dare avvio alla gestione con Scrum. Lo svolgimento dei diversi eventi del framework è formalizzato all'interno del flowchart riportato in [Figura 49-50-51].

In seguito alle attività preliminari del secondo Stage non è stata fornita alcuna rappresentazione in termini di numero di Stage e di definizione delle macro-attività contenute in essi in quanto, come ribadito precedentemente, solamente un'analisi dello specifico contesto aziendale (come ad esempio il modello di gestione adottato) e degli specifici bisogni dell'organizzazione (in termini di rigidità/flessibilità e di performance) possono fornire il corretto input alla progettazione del processo di sviluppo (all'interno del primo caso studio questo passaggio è rappresentato dalla fase di Lean Gap Analysis® del progetto di *complete lean transformation*).

5.3 Difficoltà nell'utilizzo di Scrum

A partire dall'analisi dei casi studio e da conversazioni informali con i membri dei team delle organizzazioni riportate nei casi studio, sono state individuate alcune difficoltà nell'utilizzo del framework:

Definire obiettivi raggiungibili all'interno dello Sprint. Uno dei principali standard illustrati all'interno dell'esposizione dei casi studio aziendali è lo standard di creazione di una storia. Uno dei requisiti per la creazione di una storia da inserire all'interno del Backlog è che essa sia completabile all'interno della durata di uno Sprint. La suddivisione di un deliverable non completabile nell'arco dello Sprint in sotto-deliverable raggiungibili all'interno di uno Sprint rappresenta però un aspetto di complessità per il team, soprattutto nello svolgimento dei primi Sprint. Il superamento di questa difficoltà è principalmente legato all'acquisizione di consapevolezza da parte del team attraverso lo svolgimento di più Sprint. Il tipico rischio che si corre nei primi Sprint è quello di completare poche storie rispetto a quanto pianificato, in quanto il team svolge attività che nel loro complesso non portano al completamento dei requisiti per considerare "fatte" le storie. Tale difficoltà è strettamente collegata a quella che segue.

Ragionare in termini di deliverable. Come per la precedente difficoltà, la presente si manifesta nel momento in cui il team è coinvolto nell'attività di definizione delle storie. La tendenza diffusa è quella di inserire all'interno della storia una "frase" che sta ad indicare l'attività che il membro del team andrà a svolgere. Ragionare in termini di attività non consente però di implementare i principi di ispezione e adattamento durante la riunione di Sprint Review rispetto al lavoro svolto, in quanto non rappresentano dei deliverable concreti che possono essere testati. Il team deve essere quindi supportato verso un cambio di mentalità in cui, oltre che pianificare, ragioni in termini di obiettivi testabili, ovvero vengano definiti in fase di creazione della storia anche tutti i requisiti che devono essere soddisfatti affinché tale storia si possa considerare fatta.

Ragionare in termini relativi. Precedentemente all'implementazione del framework non era prassi comune ragionare sullo sforzo richiesto dai deliverable con un approccio strutturato in modo da migliorare la pianificazione all'interno dei progetti e l'unica forma di controllo riconducibile a tale attività era la consuntivazione delle ore. Assegnare un punteggio a ciascuna storia in termini *relativi* nei confronti delle altre storie rappresenta un cambio di mentalità difficile per il team. Nella fase di introduzione al framework il team viene supportato nell'attività di assegnazione del punteggio creando delle analogie con lo sforzo richiesto in termini di giorni lavorativi e di risorse coinvolte, in quanto risulta in linguaggio più facilmente ricollegabile al loro precedente modo di fare. Successivamente però è necessario un

superamento di questa prassi, in quanto può diventare svante in fase di raffinazione e pianificazione, prediligendo un'assegnazione del punteggio come output del confronto tra le varie storie.

Risorse dedicate. Sia all'interno dell'analisi di letteratura del secondo capitolo (paragrafi 2.1 e 2.3.2) che all'interno dell'esposizione dei casi studio aziendali, emerge una difficoltà nell'applicazione del framework legata ad un orientamento prettamente verticale delle competenze aziendali il quale, a sua volta, si tramuta in un numero limitato di risorse dedicabili ai progetti, soprattutto nel caso di PMI. Il numero limitato di risorse con specifiche competenze e la richiesta di competenze sempre più specializzate nei progetti di sviluppo rappresentano un limite per i Team Scrum, che richiederebbero delle risorse dedicate.

La necessità di avere risorse dedicate al progetto nel caso in cui esse non siano coinvolte solamente nelle attività di sviluppo dello specifico progetto può essere parzialmente superata riducendo il tempo dedicato alle attività del progetto nell'arco della settimana ma allungando la durata dello Sprint (come all'interno del primo caso studio in cui la durata dello Sprint è stata fissata a due settimane, ma con al suo interno tre giorni lavorativi dedicati al progetto).

La condizione più gravosa si manifesta nel momento in cui nell'arco della giornata le risorse siano richieste in attività non previste, delle vere e proprie "emergenze" quotidiane, che creano instabilità all'interno del processo di Scrum e non permettono di portare a termine il lavoro pianificato.

Tale situazione si manifesta soprattutto nella fase iniziale di introduzione al framework e il suo superamento deve essere accompagnato da un percorso di stabilizzazione dei processi, in modo che le situazioni di "emergenza" generate dalla loro variabilità tendano asintoticamente a zero con il passare del tempo.

5.4 Adattamenti nell'utilizzo di Scrum

Come descritto nel paragrafo 5.1 attraverso il flowchart e la sua applicazione all'interno dei casi studio, la modalità adottata di implementazione del framework è in linea con l'analisi di letteratura esposta nel primo capitolo dell'elaborato. Nonostante ciò sono presenti alcuni adattamenti e considerazioni la cui analisi è oggetto del seguente paragrafo.

Incremento di prodotto. In (Sutherland & Schwaber, 2017) è riportato che: *“L'Incremento è la somma di tutti gli elementi del Product Backlog completati durante uno Sprint e del valore degli incrementi di tutti gli Sprint precedenti. Alla fine di uno Sprint, il nuovo Incremento deve risultare “Fatto”, il che significa che deve essere utilizzabile e deve soddisfare la definizione di “Fatto” data dallo Scrum Team. Un Incremento è un insieme di lavoro “Fatto” ed ispezionabile che supporta l'empirismo alla fine dello Sprint. L'Incremento è un passo verso la visione o l'obiettivo. L'Incremento deve essere utilizzabile indipendentemente dal fatto che il Product Owner decida di rilasciarlo davvero.”*

Sebbene nella letteratura di Scrum applicato al Software si faccia riferimento all'incremento di prodotto come una funzionalità che si potrebbe far testare al cliente, nel contesto dello sviluppo di prodotti tangibili risulta estremamente complesso da realizzare. Per ovviare a tale problematica, (Cooper & Sommers, 2018) esprimono il concetto di incremento di prodotto come *“risultato tangibile del lavoro completato in quello Sprint”*.

Come ribadito all'interno del paragrafo precedente, il comportamento adottato dai team prima dell'introduzione di Scrum prevedeva la focalizzazione sullo svolgimento di “attività” in orizzonti temporali non definiti e con poca importanza alla fase di pianificazione delle stesse.

L'introduzione di Scrum e l'adozione di uno standard di creazione delle storie che focalizzi il team sul ragionare in termini di deliverable concreti e testabili da completare nell'arco dello Sprint consente di avvicinarsi al concetto di incremento di prodotto potenzialmente rilasciabile espresso in letteratura. La maggior parte delle PMI non è dotata di macchinari di prototipazione rapida che consentono di adottare un approccio simile all'Extreme Manufacturing (descritto nel paragrafo 2.2) e quindi di progettare, realizzare, produrre e testare una parte del prodotto nell'arco della durata dello Sprint (anche se la durata dello Sprint fosse impostata a 4 settimane) e probabilmente non rappresenta neanche un bisogno in questo momento per le PMI italiane.

L'approccio adottato nelle realtà presentate nei casi studio aziendali è quindi quello suddividere il percorso di progettazione (come può essere quello di un “gruppo trasmissione” completo) in sotto-deliverable realizzabili all'interno di uno Sprint.

Risulta opportuno inserire un'ulteriore considerazione correlata alla tematica dell'incremento di prodotto.

Nell'analisi di letteratura del modello Hybrid Stage - Gate (paragrafo 2.3.2) viene ribadito che una delle questioni chiave risulta essere in quali fasi del processo adottare il framework Scrum. L'approccio adottato nei processi di sviluppo dei casi aziendali presentati (paragrafo 5.2) prevede l'utilizzo del framework subito dopo il kick-off del progetto, in quanto si vanno a generare tutti i requisiti affinché esso possa essere adottato (creazione del Team Scrum e del Product Backlog). Nel primo caso studio viene però riportato un esempio di adozione del framework nel primo Stage (come descritto più nel dettaglio nel paragrafo 3.5). È quindi possibile affermare che l'utilizzo del framework può essere allargato a tutte le fasi del processo di sviluppo e anche gli standard rimangono invariati. Il focus sulla generazione di deliverable concreti e testabili (coerentemente con gli standard definiti nei casi studio) permette di implementare i principi di ispezione e adattamento anche nelle prime fasi del processo. Nelle fasi più tecniche del progetto (come previsto nel paragrafo 2.3.2) l'output di ciascuno Sprint risulta però essere già pienamente allineato alla definizione dell'incremento di prodotto fornita dall'analisi di letteratura (come riportato nel secondo caso studio dove l'output dello Sprint prevedeva l'assemblaggio di diversi gruppi macchina, oltre che la progettazione di altri).

Sprint Retrospective. L'analisi di letteratura di Scrum (paragrafo 1.3.5) prevede lo svolgimento di cinque eventi nell'arco della durata dello Sprint, tra cui la riunione di Sprint Retrospective. Nell'esposizione dei casi studio aziendali non è formalmente riportato questo evento. L'attività di ispezione del processo, del team e delle relazioni in esso non viene dedicata ad una riunione a sé stante, ma è integrata all'interno della riunione di Sprint Review. Durante la riunione di Sprint Review sono discussi eventuali scostamenti nel lavoro effettuato rispetto a quanto pianificato e difficoltà riscontrate dal team; successivamente si instaura un processo di problem solving alla ricerca della causa radice del problema che può coinvolgere anche le tematiche oggetto della riunione di Retrospective. Eventuali attività di miglioramento possono essere inserite dal team all'interno del Product Backlog (come nel primo caso studio) o all'interno di un tabellone problem solving (come nel secondo caso studio). La scelta di non svolgere due eventi separati è legata principalmente ad un aspetto culturale riscontrato soprattutto nelle PMI, dove le riunioni spesso sono percepite come non a valore e quindi scoraggiate dall'imprenditore. Il problema non risiede tanto nella riunione di Scrum, ma piuttosto a come sono svolte prime dell'inizio del percorso di trasformazione. Un ulteriore motivo è l'alleggerimento del metodo in modo da favorirne un rapido apprendimento.

5.5 Conclusioni

Al momento della conclusione dell'elaborato i progetti di lean transformation dei processi di development in FILA e FreTor sono ancora in fase di attuazione. Non sono ancora disponibili misurazioni quantitative robuste da permettere un'analisi strutturata dei benefici, la quale è comunque pianificata nella fase di check del progetto.

Nonostante ciò, tramite alcune conversazioni ed interviste ai membri del team sono emersi diversi vantaggi che l'applicazione del framework ha portato, alcuni dei quali confermano quanto già evidenziato nell'analisi di letteratura:

- La formazione di team multidisciplinari prevista da Scrum garantisce un miglioramento della comunicazione interna al progetto, permettendo un migliore coordinamento e il superamento dell'impostazione di lavoro tradizionale fondata sulla divisione funzionale e i passaggi di responsabilità, che provocano sprechi all'interno del processo;
- La possibilità di vedere il lavoro che è stato fatto, il lavoro in corso e il lavoro che c'è ancora da fare consente di avere le idee più chiare di dove si sta andando, ovvero l'obiettivo da raggiungere, e di come il team sta lavorando;
- Definire degli obiettivi di breve termine che siano chiari e condivisi consente a ciascun membro del team di essere più focalizzato, di migliorare la qualità e la produttività del proprio lavoro e di integrare il miglioramento continuo all'interno di esso.

Come definito all'interno del sommario e dell'introduzione, l'obiettivo del presente elaborato consiste nel definire se l'implementazione del framework Scrum sia possibile anche nel contesto dello sviluppo di prodotti tangibili e se siano necessari degli adattamenti affinché ne sia resa possibile l'applicazione.

Nel paragrafo 5.4 sono stati presentati e analizzati gli adattamenti riscontrati nell'applicazione del framework nei due casi studio in confronto all'analisi di letteratura del primo capitolo.

L'obiettivo alla base di Scrum è abilitare al controllo empirico dei processi attraverso i principi di trasparenza, ispezione e adattamento. Ricordando ciò, è possibile affermare che l'implementazione del framework descritta nei casi studio rappresenta un caso di applicazione di Scrum nel contesto dei prodotti tangibili, in quanto i principi di trasparenza, ispezione e adattamento sono entrati a far parte del processo (implementati attraverso gli eventi, i ruoli e gli artefatti descritti).

Ciononostante, l'implementazione del framework Scrum all'interno del processo di sviluppo di prodotti tangibili deve essere supportata da un modello che operi ad un livello più elevato di gestione del processo, in modo

da garantire una visione macro e di lungo periodo. Come visto all'interno dei casi studio, la pianificazione di lungo periodo è supportata dall'integrazione delle milestones del progetto con il diagramma di Gantt.

Ritengo che, nonostante vi possano essere alcuni adattamenti nell'implementazione, l'aspetto fondamentale che deve essere tenuto in considerazione ogniqualvolta il framework debba essere applicato è l'implementazione dei principi alla base di esso, ovvero trasparenza, ispezione e adattamento.

Essi abilitano al controllo empirico, il quale consente di implementare il miglioramento continuo, il coinvolgimento di tutti i membri del team e di ridurre la variabilità del processo, con la conseguente riduzione degli sprechi generati all'interno di esso.

Bibliografia

- Agile Alliance. (2001). *Agile 101*. Tratto il giorno Maggio 14, 2020 da <https://www.agilealliance.org/agile101/>
- Agile Alliance. (2001). *Agile Manifesto*. (Ward Cunningham) Tratto il giorno Maggio 14, 2020 da <http://agilemanifesto.org>
- Architettura Modulare*. (s.d.). Tratto il giorno Giugno 03, 2020 da Wikipedia: https://it.wikipedia.org/wiki/Architettura_modulare
- Biazzo, S., & Filippini, R. (2018). *Management dell'Innovazione*. ISEDI.
- Complexity Education Project. (2018). *VUCA: Volatilità, Incertezza, Complessità e Ambiguità*. Tratto da <https://www.complexityeducation.com/2019/03/08/vuca-volatilita-incertezza-complessita-e-ambiguita/>
- Cooper, R. G. (1988). The new product process: A decision guide for managers. *Journal of Marketing Management*.
- Cooper, R. G. (2014). What's next? After Stage-Gate,. *Research-Technology Management*.
- Cooper, R. G., & Sommers, A. F. (2018). Agile-Stage-Gate for Manufacturers Changing the Way New Products Are Developed Integrating Agile project management methods into a Stage-Gate system offers both opportunities and challenges.
- Cooper, R. G., Edgett, S. J., & Kleinschmidt, E. (2002). Optimizing the Stage-Gate process: What best-practice companies do. *Research-Technology Management*.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *The Scrum Primer*.
- Morgan, J., & Liker, J. (2006). *The Toyota Production Development System*.
- Morgan, Y., & Keshta, N. (2017). Comparison between Traditional Plan-based and Agile Software Processes according to Team Size and Project Domain: A Systematic Literature Review. *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 8th IEEE Annual*.
- Project Management Institute. (2013). *PMBOK guide*.
- Puozzo, E. (2017). Modelli agile e plan-driven nello sviluppo di prodotti tangibili. Padova.
- Sammiceli, P. (2018). *Scrum per Hardware*. Leanpub.
- Schwaber, K., & Sutherland, J. (2016). *La Guida a Scrum*.
- Smith, P. G. (2007). *Flexible Product Development: Building Agility for Changing Markets*. Jossey-Bass.
- Stevenson, W. J. (2005). *Operations Management*. (Vol. 8th ed.). Londra: McGraw-Hill/Irwin.

- Sutherland, J. (2004). Agile development: Lesson Learned from the first Scrum. *Cutter Agile Project Management Advisory Service*.
- Sutherland, J. (2012). The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework.
- Sutherland, J. (2015). *Fare il doppio in metà tempo*. RIZZOLI.
- Sutherland, J. (2020, Marzo 30). *Quora - What-s-the-difference-between-Scrum-and-Agile*. Tratto il giorno Maggio 14, 2020 da <https://www.quora.com/What-s-the-difference-between-Scrum-and-Agile>
- Sutherland, J., & Schwaber, K. (2017). *The Scrum Guide™*. Tratto il giorno Aprile 2020 da <https://www.scrumguides.org/scrum-guide.html>
- Takeuchi, H., & Nonaka, I. (1986). The New New Product Development Game.
- Ullman, D. (2019). *Scrum for Hardware Design*. David G. Ullman LLC.
- Vedsmand, T. (2016). *Integrating Agile with Stage-Gate® - How New Agile-Scrum Methods Lead to Faster and Better Innovation*. Tratto il giorno Giugno 14, 2020 da Innovation Management: <https://innovationmanagement.se/2016/08/09/integrating-agile-with-stage-gate/>
- Zhang, L. (2012). Managing project changes: Case studies on stage iteration and functional interaction. *International Journal of Project Management*.

Indice delle Figure

<i>Figura 1 - Riassunto grafico del funzionamento di Scrum. Fonte: (Sutherland, 2012)</i>	<i>12</i>
<i>Figura 2 - Esempio di Product Backlog. Fonte: (Sutherland, 2012).....</i>	<i>20</i>
<i>Figura 3 - Esempio Sprint Backlog. Fonte: (Sutherland, 2012).....</i>	<i>20</i>
<i>Figura 4 - Cambiamento di requisiti nelle varie iterazioni. Fonte: (Smith, 2007).....</i>	<i>21</i>
<i>Figura 5 - Esempio Scrum Board. Fonte: auxiell.....</i>	<i>22</i>
<i>Figura 6 - A sinistra, esempio di Sprint Burn-down Chart. Fonte: (Sutherland, 2012) Figura</i>	
<i>7 - A destra, esempio Sprint Burn-up Chart.....</i>	<i>23</i>
<i>Figura 8 - Hybrid Agile Stage Gate (Cooper, et al., 2018).....</i>	<i>38</i>
<i>Figura 9 - auxiell lean system.....</i>	<i>46</i>
<i>Figura 10 - 4D e ciclo SPDCa - https://www.auxiell.com</i>	<i>47</i>
<i>Figura 11 - Value Proposition di auxiell - https://www.auxiell.com</i>	<i>47</i>
<i>Figura 12 - Logo FILA Industria Chimica s.p.a. - www.filasolutions.com.....</i>	<i>48</i>
<i>Figura 13 - Centro di ricerca e sviluppo di FILA.....</i>	<i>49</i>
<i>Figura 14 - Current State Map e Muda Identification del processo di sviluppo di FILA -</i>	
<i>Fonte: auxiell.....</i>	<i>53</i>
<i>Figura 15 - Muda Analysis Current State Map FILA - Fonte: auxiell.....</i>	<i>56</i>
<i>Figura 16 - Rappresentazione Future State Map FILA - Fonte: auxiell.....</i>	<i>60</i>
<i>Figura 17 - Definizione linee guida del processo di pianificazione in FILA - Fonte: auxiell</i>	
<i>.....</i>	<i>62</i>
<i>Figura 18 - Rappresentazione dell'organizzazione del processo di development in FILA</i>	<i>63</i>
<i>Figura 19 - Standard FILA definizione storia da raffinare - Fonte: auxiell.....</i>	<i>67</i>
<i>Figura 20 - Product Backlog - Fonte: auxiell.....</i>	<i>67</i>
<i>Figura 21 - esempio storia da raffinare e storia raffinata - Fonte: auxiell.....</i>	<i>68</i>
<i>Figura 22 - standard FILA storia pronta - Fonte: auxiell</i>	<i>69</i>
<i>Figura 23 - Sequenza di selezione delle storie dal Backlog</i>	<i>72</i>
<i>Figura 24 - Tabellone dello Sprint - Fonte: auxiell.....</i>	<i>72</i>
<i>Figura 25 - Tabellone dei Task - Fonte: auxiell.....</i>	<i>73</i>
<i>Figura 26 - Grafico storico di velocità degli Sprint - Fonte: auxiell.....</i>	<i>76</i>
<i>Figura 27 - Storie compilate - Fonte: auxiell.....</i>	<i>77</i>
<i>Figura 28 - Backlog virtuale con le storie da raffinare - Fonte: auxiell.....</i>	<i>78</i>
<i>Figura 29 - Storia raffinata - Fonte: auxiell.....</i>	<i>78</i>
<i>Figura 30 - Product Backlog in FILA.....</i>	<i>79</i>
<i>Figura 31 - Tabellone dello Sprint del progetto in FILA</i>	<i>80</i>
<i>Figura 32 - Tabellone dei Task in FILA.....</i>	<i>81</i>
<i>Figura 33 - Logo FreTor s.r.l. - Fonte: https://www.fretor.com</i>	<i>84</i>
<i>Figura 34 - Rappresentazione Future State Map FreTor con aggiunta del dettaglio degli</i>	
<i>Sprint- Fonte: auxiell.....</i>	<i>87</i>
<i>Figura 35 - Standard compilazione requisito - Fonte: auxiell</i>	<i>88</i>
<i>Figura 36 - Standard tipologie di requisiti - Fonte: auxiell.....</i>	<i>89</i>
<i>Figura 37 - Standard di classificazione e posizionamento dei requisiti - Fonte: auxiell.....</i>	<i>90</i>
<i>Figura 38 - Standard di creazione delle storie - Fonte: auxiell.....</i>	<i>91</i>
<i>Figura 39 - Esempio tabellone dei requisiti - Fonte: FreTor</i>	<i>91</i>
<i>Figura 40 - Tabelloni Visual adottati in FreTor in cui è evidenziato il Product Backlog -</i>	
<i>Fonte: FreTor.....</i>	<i>92</i>
<i>Figura 41 - Esempio Storie selezionate in FreTor - Fonte: FreTor.....</i>	<i>93</i>
<i>Figura 42 - Modulo di reportistica del Daily Scrum - Fonte: FreTor.....</i>	<i>93</i>
<i>Figura 43 - Tabella delle traslazioni per la commessa in analisi- Fonte: FreTor.....</i>	<i>94</i>
<i>Figura 44 - Tabellone dello Sprint ad inizio Sprint Review - Fonte: FreTor.....</i>	<i>95</i>

<i>Figura 45 - Grafico di velocità degli Sprint - Fonte: FreTor</i>	<i>95</i>
<i>Figura 46 - Tabellone dello Sprint al termine della seconda riunione di Sprint Planning - Fonte: FreTor</i>	<i>97</i>
<i>Figura 47 - Grafico di velocità dello Sprint aggiornato al secondo Sprint descritto - Fonte: FreTor</i>	<i>97</i>
<i>Figura 48 - Esempio di suddivisione delle storie.....</i>	<i>98</i>
<i>Figura 49 - Flowchart utilizzo framework Scrum - Fonte: auxiell</i>	<i>101</i>
<i>Figura 50 - Flowchart utilizzo framework Scrum - Fonte: auxiell</i>	<i>103</i>
<i>Figura 51 - Flowchart utilizzo framework Scrum - Fonte: auxiell</i>	<i>104</i>
<i>Figura 52 - Modello di processo proposto per l'implementazione di Hybrid Stage-Gate</i>	<i>105</i>