

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE

DEI SISTEMI INDUSTRIALI

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA GESTIONALE

TESI DI LAUREA

Approccio Agile alla gestione dei progetti in
ambito manifatturiero - Il caso Galdi srl

Relatore: Prof.ssa Chiara Verbano

Candidato: Angelica Pozzobon

1159849

Anno Accademico: 2018/19

Ringraziamenti

Desidero ringraziare il relatore di questa tesi, la professoressa Chiara Verbano, che mi ha dato l'opportunità di svolgere questo lavoro relativo ad un argomento interessante, donandomi consigli preziosi e supporto durante il tirocinio e la stesura dell'elaborato.

Un profondo ringraziamento va anche a Galdi srl. In particolare a Federico, per l'aiuto a comprendere la realtà aziendale, a Michele M., per tutti i consigli e gli incoraggiamenti dei primi mesi di tirocinio, a Filippo, Benedetta, Paolo e a tutto il team che ha lavorato nella realizzazione del progetto applicativo di questa tesi: grazie per l'aiuto, il supporto, i suggerimenti e il tempo dedicatomi anche quando le cose da fare erano molte.

Un ringraziamento speciale va ad Alice, che mi ha insegnato tante cose, e a Federica e Maria, che stanno continuando a insegnarmene molte.

Ringrazio inoltre la mia famiglia. Senza il loro supporto e sostegno in tutte le forme possibili non sarei potuta arrivare dove sono adesso. Grazie mamma, per avermi incoraggiata sempre, per aver sopportato i momenti di sconforto e per aver sempre creduto che potessi farcela.

Infine, grazie Nicola. Grazie per esserci stato sempre: a condividere i momenti di gioia, a risollevarmi nei momenti di tristezza, a salvarmi nei momenti di difficoltà, a prendermi in giro quando mi stavo prendendo troppo sul serio.

Neanche io senza di te sarei diventata chi sono ora.

Indice

Introduzione	1
1 Obiettivi e metodologie di ricerca	3
2 L'approccio <i>Agile</i> al <i>Project Management</i>	7
2.1 L'approccio <i>Agile</i> nel mondo software	8
2.1.1 La nascita dell'approccio	9
2.1.2 I modelli tradizionali	10
2.1.3 Il confronto tra metodologie tradizionali e <i>Agile</i>	12
2.1.4 Le metodologie agili	13
2.2 L'approccio <i>agile</i> in altri settori	21
2.3 Analisi delle implicazioni <i>agile</i> nella gestione dei progetti	23
2.3.1 Le aree di conoscenza nella gestione dei progetti	23
2.3.2 L'impatto <i>agile</i> nelle aree di conoscenza	26
3 Il caso Galdi	35
3.1 L'azienda	35
3.1.1 La storia	35
3.1.2 Il prodotto	36
3.1.3 Il principio di funzionamento del prodotto realizzato	37
3.1.4 Il processo produttivo	38
3.2 La gestione dei progetti di sviluppo nuovi prodotti in Galdi	40
3.2.1 Classificazione progetti di sviluppo nuovi prodotti	41
3.2.2 Nascita di un progetto	42
3.2.3 Esecuzione del progetto	46
3.2.4 Elementi di forza e di debolezza del processo	48
4 L'adozione di Scrum e il progetto applicativo	51
4.1 Il progetto di miglioramento: obiettivi e strumenti	51
4.1.1 Il <i>Product Canvas</i>	51
4.1.2 L'uso di Scrum	53

4.2	Esempio applicativo: il progetto Dry Food	55
4.2.1	La storia del progetto e le sue sfide	56
4.2.2	Analisi del progetto	59
4.2.3	Analisi elementi <i>agile</i>	71
4.2.4	Analisi implementazione Scrum	76
4.3	Valutazione dell'approccio	79
5	Conclusioni	83
	Elenco delle figure	87
	Riferimenti bibliografici	89

Introduzione

Negli ultimi anni il progresso tecnologico è avvenuto a una velocità incredibile e il ciclo vita dei prodotti si è ridotto notevolmente. L'innovazione ha assunto un ruolo ancor più rilevante rispetto al passato e la necessità di introdurre nel mercato nuovi prodotti in tempi ristretti è aumentata.

Insieme a ciò si è sviluppata una maggior consapevolezza da parte dei clienti, che cercano prodotti in grado di rispondere alle loro esigenze, hanno la possibilità di informarsi maggiormente riguardo cosa offre il mercato grazie all'utilizzo delle tecnologie e per questo risultano essere più esigenti nelle loro richieste.

Per questi motivi le aziende hanno cercato nuovi metodi con cui approcciare i progetti di sviluppo nuovi prodotti, che consentano di essere più veloci nello sviluppo e preparati a fronteggiare i cambiamenti. Una delle soluzioni è stata l'adozione degli approcci *Agile*.

Con *Agile* si intendono una serie di approcci nati nel mondo dell'*Information Technology* con l'obiettivo di rispondere al cambiamento in maniera rapida ed efficace, creando un prodotto di qualità e aumentando la soddisfazione del cliente e del team stesso che lavora al progetto.

Viste le premesse desiderabili e i risultati positivi ottenuti nell'industria software, il tema ha cominciato ad interessare molti altri settori, dall'istruzione ai servizi, dalle società di consulenza fino all'industria manifatturiera e molte aziende stanno cercando di adottarne gli elementi chiave per assicurarsi un vantaggio competitivo.

Un ambito applicativo interessante risulta essere lo sviluppo di nuovi prodotti, in cui molti degli elementi descritti possono portare beneficio e trovare applicazione, sebbene siano necessari opportuni accorgimenti. In relazione a ciò è possibile valutare l'impatto delle metodologie nella gestione stessa dei progetti, che dovrà essere anch'essa adattata ai nuovi metodi, in modo da creare coerenza rispetto alla progettazione.

In questa tesi viene valutata l'applicazione dei suddetti approcci nella gestione dei progetti di sviluppo prodotto in contesti diversi rispetto al mondo software, in particolare viene analizzato il caso di una media impresa appartenente all'industria manifatturiera, valutando l'efficacia della proposta e i risultati ottenuti.

La tesi si struttura nei capitoli di seguito descritti:

- nel primo capitolo vengono brevemente discussi obiettivi e metodologie utilizzati nella stesura della tesi;
- nel secondo capitolo viene presentato l'approccio *agile* a partire dalla sua nascita e dalle implementazioni nel mondo software fino alle sue applicazioni in contesti differenti. Infine viene analizzato come l'approccio *agile* influenzi il *project managment*;
- nel terzo capitolo viene presentata l'azienda analizzata evidenziandone il contesto in cui opera e le metodologie di gestione dei progetti finora utilizzate;
- nel quarto capitolo vengono presentati gli obiettivi di miglioramento nel processo di sviluppo nuovi prodotti dell'azienda, viene illustrato il progetto utilizzato come esempio applicativo e vengono discussi i benefici e le criticità riscontrate nell'applicazione di tale approccio;
- nel quinto ed ultimo capitolo vengono brevemente ripresi gli obiettivi di progetto, riassunti i risultati ottenuti e discussi utilità e limitazioni dell'elaborato.

1 | Obiettivi e metodologie di ricerca

L'obiettivo di questa tesi è quello di valutare la possibilità di applicare un approccio ispirato alla filosofia *agile* e alla metodologia *scrum* alla gestione di progetti in contesti diversi dallo sviluppo software, in particolare alla gestione di progetti di sviluppo nuovi prodotti di una media impresa operante su commessa nel settore manifatturiero, evidenziando vantaggi e criticità riscontrati con l'utilizzo dell'approccio.

È stata in primo luogo svolta un'analisi della letteratura al fine di conoscere lo stato dell'arte riguardo l'argomento trattato in questa tesi. Il focus è stato nella comprensione del fenomeno *agile* nel mondo *IT* e delle sue applicazioni in tale contesto. Successivamente la ricerca è stata estesa a settori industriali diversi dall'*IT*. L'attenzione è stata posta in maniera particolare sull'impatto degli approcci nel *project management* e le implicazioni che l'adozione di tali approcci hanno sul modo di gestire i progetti.

La ricerca è avvenuta tramite la consultazione di database quali *Web of Science* e *Scopus* ed è stata successivamente estesa a *Google Scholar*. Le parole chiave utilizzate sono state: *Agile Project Management*, *agile benefits*, *agile challenges*, *agile hybrid approach*, *agile New Product Development*. Una volta individuati alcuni articoli considerati attinenti all'argomento preso in esame in questa tesi, la ricerca è stata estesa consultando anche le fonti utilizzate in tali articoli. In Tabella 1 vengono riportati gli articoli individuati.

Sono stati inoltre consultati siti web e volumi di riferimento per l'ambito di interesse: i primi per avere un'idea generale dell'argomento di studio, i secondi per approfondire alcuni punti specifici. Nel momento in cui la ricerca è stata estesa ad altri settori non sono stati trovati molti articoli a cui fare riferimento, a causa della tematica di ricerca ancora poco esplorata.

In seconda battuta è stato analizzato il caso aziendale. L'azienda è una media impresa del territorio veneto, che opera in ambito manifatturiero, producendo su commessa grandi macchine automatiche. Il mercato di riferimento è globale. L'azienda aveva degli obiettivi specifici di miglioramento mirati a risolvere alcune criticità nella gestione dei progetti di sviluppo nuovi prodotti emerse in casi precedenti. La soluzione scelta è stata quella di adottare alcuni strumenti *agile* e in particolare la metodologia Scrum ai progetti di

Articolo	Autore	Anno
New directions on agile methods: a comparative analysis	Abrahamsson et al.	2003
A comparison between agile and traditional software development methodologies	Awad	2005
Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry	Batra et al.	2010
Extreme Programming explained: Embrace change	Beck	2004
Comparison between traditional plan-based and agile software processes according to team size & project domain (A systematic literature review)	Keshta	2017
Challenges of migrating to agile methodologies	Nerur	2005
Agile project management in new product development and innovation processes: challenges and benefits beyond software domain	Ciric et al.	2018
Improved product development performance through Agile/Stage-Gate hybrids: The next-generation Stage-Gate process?	Sommer et al.	2015
Can agile project management be adopted by industries other than software development?	Conforto et al.	2014
Benefits of agile project management in a non-software development context-A literature review	Gustavsson	2016

Tabella 1: Articoli rilevati dalla ricerca

Fonte: propria elaborazione

sviluppo. L'azienda ha deciso di cominciare l'implementazione di questo metodo in due progetti pilota, uno dei quali è stato riportato in questa tesi.

Obiettivo di questa tesi è valutare la fattibilità, i benefici e le criticità riscontrate nell'implementazione di questa metodologia.

La raccolta dei dati riguardo l'azienda e il progetto è avvenuta nel contesto di uno stage della durata di cinque mesi, in tre dei quali la presenza in azienda è stata a tempo pieno, mentre negli ultimi due a tempo parziale.

In primo luogo è stata compresa la realtà operativa dell'azienda e analizzata la gestione precedente dei progetti di sviluppo nuovi prodotti. La raccolta di questi dati è avvenuta tramite consultazione della documentazione aziendale e l'interazione con figure chiave aziendali. In seguito si è osservato lo svolgimento della fase di sviluppo di uno dei progetti pilota che l'azienda stava portando avanti, partecipando agli incontri relativi e svolgendo interviste, strutturate e non, ai soggetti coinvolti. Un elemento interessante è stata la possibilità di svolgere una visita presso la sede del cliente finale, per raccogliere dati

riguardo i suoi feedback e valutare direttamente l'impatto di questa gestione percepito dal committente.

Una parte del progetto è stata poi seguita nei successivi tre mesi, in maniera meno focalizzata, mentre si svolgevano altre attività all'interno dell'azienda, in questo modo è stato possibile valutare gli impatti dell'uso dell'approccio *agile* in progettazione anche nelle fasi successive del progetto.

L'elaborazione dei benefici e delle criticità emersi e osservati è stata svolta tenendo in considerazione quanto precedentemente emerso dall'analisi della letteratura.

2 | L'approccio *Agile* al *Project Management*

Con il termine *Agile Project Management* si intende un metodo iterativo e incrementale di gestire le attività di pianificazione, progettazione, sviluppo, costruzione e test di un progetto.

Nell'*Agile practice guide* (PMI, 2017), l'*Agile Project Management* viene definito come l'insieme di metodi e sistemi utili a gestire un progetto con un ciclo di vita agile, ovvero iterativo e incrementale, caratterizzato da requisiti dinamici, ripetizione delle attività e frequenti interazioni con il cliente che consentano di ricevere dei feedback anche su sottoinsiemi funzionanti del prodotto finito, prima che sia completo.

Nato dal mondo software, visti i benefici e la necessità di fronteggiare il cambiamento e di avere *time to market* molto brevi, l'approccio *agile* sta interessando e trovando sempre maggior impiego anche in altri settori, come quello manifatturiero. Tuttavia, viste le caratteristiche del contesto, la sua trasposizione e implementazione non risulta essere automatica. Le ricerche in merito sono ancora recenti e stanno avendo luogo in questi anni.

Le pratiche¹ individuate dall'*agile project management* includono (Conforto et al., 2014):

- Uso del concetto di *product vision* e di strumenti di comunicazione della pianificazione semplici.

Il concetto di visione prevede l'uso di strumenti *visual*, come *visual board*, post-it, figure e disegni per fornire una semplice descrizione del sistema generale e richiede un alto grado di interazione tra i membri del team e una forte collaborazione coi i clienti e gli *stakeholder*² chiave. L'obiettivo è focalizzare l'attenzione sulle sfide e i problemi principali che il team deve affrontare nei progetti e aiutare a identificare quali aspetti chiave rappresentano il maggior valore per il cliente e il mercato. Utilizza metafore,

¹Pratiche: Una specifica azione professionale o di gestione che contribuisce all'esecuzione di un processo e che potrebbe comprendere una o più tecniche e strumenti.

²*Stakeholder*: ciascuno dei soggetti direttamente o indirettamente coinvolti in un progetto o nell'attività di un'azienda

figure e prototipi e ciò si differenzia dal classico modo di rappresentare l'ambito di progetto, solitamente espresso sotto forma di *work breakdown structure*³ o distinta base.

- Uso di una pianificazione iterativa.

Altro elemento caratterizzante è la pianificazione delle attività, che non avviene in un unico momento all'inizio del progetto, ma viene ripetuta più volte lungo la sua vita. Ciò per essere in linea con lo sviluppo incrementale del prodotto, ossia la sua suddivisione in sotto gruppi funzionanti sviluppati e pianificati uno dopo l'altro, e per poter facilitare l'inclusione di cambiamenti.

- Maggior autonomia del team di progetto.

Le attività di sviluppo sono gestite e organizzate in maniera autonoma dal team, in modo da favorire il coinvolgimento e l'interesse nei confronti del progetto. Anche le attività di monitoraggio e aggiornamento del piano di progetto sono svolte in autonomia dal team stesso in modo da migliorare le interazioni e l'efficacia delle comunicazioni all'interno del gruppo. Ciò consente inoltre di far crescere professionisti che sappiano gestire e fronteggiare progetti complessi (Winter et al., 2006).

- Monitoraggio e aggiornamento del piano di progetto svolti frequentemente.

Le attività di monitoraggio e aggiornamento del progetto vengono svolte al termine di ogni iterazione (indicativamente non maggiori di quattro settimane), non essendo pianificato tutto all'inizio. Ciò si differenzia dal *project management* classico in cui le ripianificazioni vengono svolte solo in caso di scostamenti dalla situazione inizialmente prevista.

2.1 L'approccio Agile nel mondo software

Agile è la capacità di evolvere e rispondere al cambiamento in modo da avere successo in un ambiente turbolento e incerto (<https://www.agilealliance.org>).

³*Work Breakdown Structure*: strumento per la scomposizione analitica di un progetto. Attraverso un diagramma, o mediante elenchi strutturati e descrittivi, essa mostra tutte le parti di un progetto a diversi livelli di dettaglio, dai primi sotto-obiettivi fino ai compiti specifici.

Il termine è utilizzato per raggruppare una serie di metodologie e di *best practices* utilizzate nello sviluppo software, che si focalizzano sulla collaborazione stretta tra il team di sviluppo e gli *stakeholder* economici, la consegna di valore continuo al cliente e la capacità del team di auto-organizzarsi.

2.1.1 La nascita dell'approccio

Nel 2001 un gruppo composto da diciassette sviluppatori software si riunisce nello Utah in un contesto informale (un soggiorno in una località sciistica) per discutere le idee e gli approcci emersi a partire dalla seconda metà degli anni '50 nello sviluppo software, nati in contrapposizione al metodo tradizionale, a cui ci si riferisce comunemente con il termine *waterfall*.

Queste metodologie erano diverse tra loro, ma durante la discussione fu possibile individuare dei punti comuni. L'unione di questi valori condivisi, fu espressa nel "Manifesto per lo Sviluppo *Agile* di Software" e nei corrispondenti dodici principi, di seguito riportati.

Manifesto per lo Sviluppo *Agile* di Software

Stiamo scoprendo metodi migliori di creare software, sviluppandolo e aiutando gli altri a fare lo stesso. Grazie a questa attività siamo arrivati a considerare importanti:

Individui e interazioni più che processi e strumenti

Il software funzionante più che la documentazione esaustiva

La collaborazione con il cliente più che la negoziazione dei contratti

Rispondere al cambiamento più che seguire un piano

Detto ciò, fermo restando il valore degli elementi a destra, consideriamo più importanti le voci a sinistra.

I principi sottostanti il manifesto

1. La nostra massima priorità è soddisfare il cliente rilasciando fin da subito e in maniera continua software di valore. lo sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.
2. Accogliamo i cambiamenti nei requisiti, anche durante stadi avanzati del-
3. Consegniamo frequentemente software funzionante, con cadenza varia-

- bile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.
4. Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.
 5. Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.
 6. Il modo più efficiente e più efficace per comunicare con il team e all'interno del team è la conversazione faccia a faccia.
 7. Il software funzionante è il principale metro di misura del progresso.
 8. I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante.
 9. La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità.
 10. La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.
 11. Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano.
 12. A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

Fonte: <https://agilemanifesto.org/>

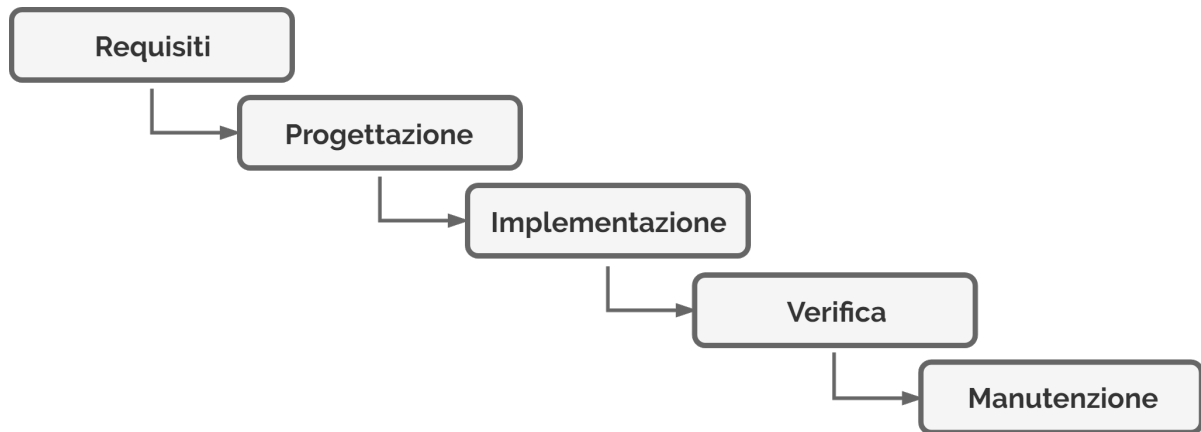
Il termine *agile* fu scelto per evidenziare la capacità di adattarsi e rispondere al cambiamento.

Sebbene ogni metodologia ponga un'importanza diversa a questi principi, favorendone alcuni anziché altri, essi rappresentano dei punti in comuni a tutte. È importante comprenderli, perché costituiscono gli elementi fondanti su cui fare affidamento.

2.1.2 I modelli tradizionali

I modelli tradizionali di sviluppo software, o *plan-based design*, prevedono che il processo di realizzazione sia strutturato secondo una sequenza lineare di fasi.

Il primo modello sviluppato è stato il modello a cascata, *waterfall*, le cui fasi, che seguono il flusso dall'alto al basso e non permettono di poter tornare alle fasi precedenti, come una cascata appunto, sono riportate di seguito in Figura 2.1.

Figura 2.1: Struttura generale del modello di sviluppo *waterfall*

Fonte: Propria elaborazione

Il processo di sviluppo è diviso in fasi sequenziali, ognuna di esse viene documentata e produce un output che sarà poi utilizzato nella fase successiva.

Le fasi sono:

- **Analisi dei requisiti:** ha lo scopo di determinare cosa farà il sistema, potrebbe comprendere o essere preceduta da uno studio di fattibilità. Genera come output un documento che specifica i requisiti di prodotto.
- **Progettazione:** i requisiti vengono studiati e analizzati allo scopo di determinare come il sistema svolgerà le funzioni definite. Viene generata l'architettura di massima, la suddivisione in moduli e le loro caratteristiche e relazioni.
- **Implementazione:** scrittura del codice dei vari moduli e loro integrazione. Quanto pensato nella fase precedente viene concretizzato nella realizzazione del software come output.
- **Verifica:** vengono svolti i test sistematici del software e le operazioni di *debug* per verificare la correttezza dell'intero sistema. Potrebbe essere prevista una fase di test finale con qualche utente specifico, prima della *release* del software al pubblico generale.
- **Manutenzione:** segue la consegna del prodotto al cliente e comprende tutte le attività volte a migliorare e correggere il sistema nel tempo.

Il modello riprende la sequenza di passi tipica dell'industria manifatturiera e fu il primo modello ad essere applicato per superare la codifica *code and fix*⁴, utilizzata in precedenza.

Dal modello a cascata si sono sviluppate una serie di altre metodologie che hanno le seguenti caratteristiche comuni:

- **Approccio predittivo:** queste metodologie si propongono di pianificare e definire una grande parte del software da sviluppare con un elevato grado di dettaglio prima di passare allo sviluppo vero e proprio. Oltre a ciò, viene pianificato chi eseguirà i compiti, le tempistiche e il budget.
- **Documentazione completa:** la raccolta dei requisiti e la loro documentazione ha un ruolo importante nelle metodologie tradizionali. È prevista una raccolta di tutte le specifiche in anticipo, prima di cominciare la fase di codifica, in modo da poter pianificare come realizzare poi il codice. Talvolta i requisiti potrebbero non essere definiti dal team di sviluppo software, per cui la documentazione deve essere estremamente accurata ed esaustiva.
- **Orientamento ai processi:** una grande importanza viene data ai processi e alla definizione di fasi di sviluppo ben definite, che possano essere seguite da tutti allo stesso modo. Vengono definite procedure che devono essere seguite dal team, dai manager e in generale da tutti i soggetti coinvolti.
- **Importanza degli strumenti:** all'inizio dell'implementazione dei metodi tradizionali è presente una fase di scelta degli strumenti che si utilizzeranno, che si tratti di strumenti di *project management*, linguaggi di programmazione, *editor* di codice, compilatori o quanto altro necessario.

2.1.3 Il confronto tra metodologie tradizionali e Agile

Le differenze tra le metodologie tradizionali e agili vengono riassunte nella tabella seguente:

⁴Il *code and fix* è un approccio quasi totalmente privo di organizzazione. Il metodo prevede di capire approssimativamente quale sarà la risposta finale del software e di provare ripetutamente a generare codice correggendo gli errori.

Categorie	Tradizionali	Agile
Modello di sviluppo	Sequenziale	Iterativo e/o incrementale
Focus	Processi	Persone
Stile di gestione	Autoritario e centralizzato	Collaborativo e facilitativo
Assegnazione dei compiti	Ognuno ha un proprio compito definito, favorisce la specializzazione	I team si auto organizzano, favorisce una conoscenza ampia
Ruolo del cliente	Importante, presente nella definizione dei requisiti e nelle fasi finali	Critico, presente (o rappresentato) durante tutto l'avanzamento
Documentazione	Diffusa e completa	Presente quando necessario

Fonte: tratto da Nerur et al. (2005)

Le metodologie tradizionali prevedono una serie di processi sistematici, organizzati, pianificabili e efficienti per lo sviluppo di software. Tuttavia si dimostrano molto rigorose, lente e inefficienti in progetti esplorativi, in cui i requisiti non sono stabili, ma in evoluzione, che preferirebbero metodi di progettazione creativi anziché sistematici (Keshta, 2017).

2.1.4 Le metodologie agili

Le metodologie *agile* sono molte, tra cui *Scrum*, *Crystal*, *Extreme Programming*, *Adaptive Software Development*, *Feature Driven Development* e *Lean Software Development*. Ogni metodo ha vantaggi differenti ed è formato da un suo specifico set di strumenti e pratiche sottostanti. La maggior parte di essi non è stato pensato e non considera aspetti di *Project Management*, ma si limita a fornire strumenti per guidare il processo di sviluppo. In questa sezione vengono riportate alcune delle metodologie più comuni per evidenziare come si concretizzano i valori e i principi trattati nel paragrafo precedente in esempi pratici, prestando particolare attenzione alle due più diffuse, XP e Scrum.

2.1.4.1 Extreme Programming (XP)

L'*Extreme Programming* (XP) è una metodologia che prevede una serie di pratiche tecniche che puntano alla scrittura di codice di qualità e alla rapidità di risposta ai cambiamenti di requisiti. L'approccio prende il nome dal fatto che va ad estremizzare alcune pratiche comuni considerate di buon senso nella programmazione. Ad esempio, vista l'importanza di testare il software una volta scritto, vengono preparati test automatici ancor prima di scrivere il codice che si intende testare.

Come le altre metodologie *agile*, è caratterizzata da cicli di sviluppo brevi, una pianificazione incrementale e feedback continui. (Beck, 2004 e www.devx.com)

Le regole dell'*Extreme Programming* sono riassunte di seguito:

- *Planning Game*: è una riunione di pianificazione che avviene una volta per iterazione, tramite la quale i programmatori stabiliscono lo sforzo richiesto, spesso in termini di un'apposita unità di misura detta velocità, per svolgere i vari compiti e il cliente stabilisce ambito e tempistiche delle *release*.
- *Small Releases*: la consegna del software avviene tramite *release* piccole e frequenti, in modo da velocizzare il *time to market* e ottenere feedback più velocemente possibile.
- *Metaphor*: il sistema dovrebbe essere descritto, se possibile, tramite una metafora. Questa può essere una storia, detta *user story*, che viene utilizzata da clienti, programmatori e manager per raccontare il funzionamento del sistema.
- *Simple Design*: l'enfasi è nel programmare la soluzione più semplice possibile per realizzare la funzionalità che si intende implementare. È importante non sviluppare cose che non saranno mai utilizzate e rimuovere il codice extra.
- *Test driven development*: tutto va testato, cercando di automatizzare i test il più possibile.
- *Refactoring*: anziché progettare l'intero sistema a monte, il sistema viene pianificato mano a mano che si procede, apportando i miglioramenti quando necessario. In particolare, i cambiamenti al codice devono venire eseguiti senza modificare le

funzionalità esterne ma applicando modifiche all'architettura al fine di renderla più semplice e funzionale.

- *Pair Programming*: programmare avviene in coppia, con due programmatori nella stessa postazione. Ciò consente la discussione in tempo reale e migliora l'attenzione ai requisiti, alla progettazione, ai test e alla programmazione stessa.
- *Collective Code Ownership*: ognuno è responsabile per tutto il codice e ogni membro del team può apportare modifiche al codice in ogni momento.
- *Continuous Integration*: il nuovo codice viene integrato con quanto già presente non appena è pronto. Il sistema viene costruito ogni volta e ritestato in maniera automatica. Questo eviterà ritardi nelle fasi successive, dovuti a problemi di integrazione.
- *Sustainable Pace*: idealmente i membri del team non dovrebbero lavorare più di quaranta ore settimanali per raggiungere le scadenze di progetto. Questo consente delle consegne più consistenti, prevedibili e ripetibili.
- *Whole Team*: il team funziona come un intero. I membri sono incoraggiati ad avere una conoscenza generale anziché essere specializzati, a conoscere tutte le tecnologie e tutti i requisiti di progetto.
- *Coding Standards*: al fine di massimizzare la comunicazione, gli standard di scrittura del codice sono definiti dal team, allo scopo di mantenere una struttura coerente.
- *Onsite Customer*: dev'essere possibile avere accesso diretto e costante all'utente che utilizzerà il software, in modo da lavorare il più velocemente possibile.

Il team

I ruoli in un team XP sono: il cliente, i programmatori, il coach, che guida i membri del team nei processi e nelle tecniche pratiche dell'XP, e il *tracker*, che monitora i progressi del team e avverte quando sono necessarie modifiche alla pianificazione o ai task assegnati a un'iterazione.

Fasi di un progetto XP

Il ciclo vita di un progetto XP è diviso in sei fasi, illustrate in Figura 2.2 e di seguito riportate:

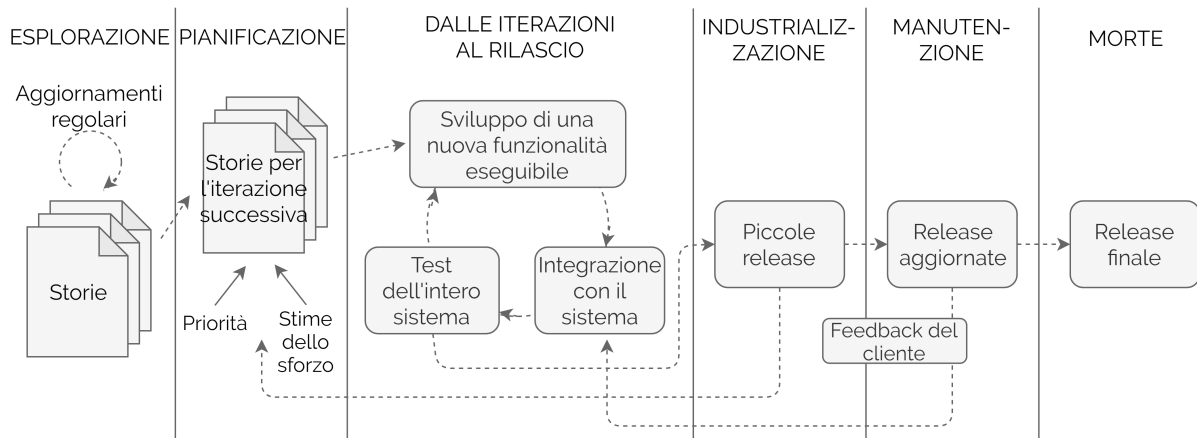


Figura 2.2: Ciclo di vita del processo XP

Fonte: adattato da Abrahamsson et al., 2003

- Esplorazione: il cliente scrive le storie utente e assegna loro una priorità. Il team di programmatori stabilisce la velocità con la quale le storie possono essere completate.
- Pianificazione delle *release*: il cliente sceglie le storie da sviluppare durante la *release* sulla base delle funzionalità o della data di consegna desiderata. Una volta scelta la storia, questa viene suddivisa in task dagli sviluppatori. I programmatori si assumono una responsabilità sui task e stimano quelli che li coinvolgono. Il carico di lavoro viene comparato con le loro performance storiche per fare in modo che non ci sia un carico eccessivo su qualcuno e che sia bilanciato tra i vari membri.
- Dalle iterazioni alla *release*: durante la prima iterazione il team di sviluppo crea l'architettura dell'intero sistema e continua poi a integrare e testare il codice, completando i task. Periodicamente un membro del team controlla i progressi e comunica queste informazioni agli altri in modo da poter organizzare eventuali modifiche.
- Industrializzazione: durante questa fase vengono svolti ulteriori test e controlli del sistema prima che il prodotto sia rilasciato al cliente.
- Manutenzione: ulteriori idee e suggerimenti vengono documentati per un'implementazione successiva.
- Morte: nell'ultima fase infine viene compilata tutta la documentazione necessaria e il sistema non avrà altre modifiche all'architettura, al design o al codice. Il cliente non ha nessun'altra storia da implementare.

2.1.4.2 Scrum

Scrum è un *framework* per la gestione dei progetti. Non definisce, quindi, metodologie e pratiche specifiche per lo sviluppo software e per questo si integra bene con altri approcci. L'obiettivo principale è quello di consegnare al cliente dopo ogni iterazione un incremento di valore del software. Il termine *scrum* deriva dal nome di una fase del rugby, mischia in italiano, il cui scopo è di far ripartire il gioco con una lotta per il possesso del pallone dopo un'infrazione minore o un'interruzione. (Schwaber, 2017 e Schwaber, 2002)

Il team

Elemento fondamentale della metodologia scrum sono le persone. Lo *scrum team* deve essere interfunzionale e in grado di organizzarsi da sè. Deve cioè avere tutte le competenze necessarie a realizzare il progetto senza dipendere da altri e deve essere il team stesso a decidere come realizzare il lavoro. I soggetti del team scrum sono elencati di seguito:

- **Product Owner:** il ruolo del *product owner* è quello di rappresentare il cliente e i suoi interessi.
- **Development team:** è formato da coloro che lavorano per realizzare il prodotto. Viene consigliato un numero di membri che varia da tre a nove. Un team troppo piccolo non consentirebbe sufficienti interazioni e diminuirebbe la produttività, troppo numeroso comporterebbe invece uno sforzo eccessivo per la coordinazione dei membri.
- **Scrum Master:** può essere considerato il *project manager* in scrum e il suo ruolo è quello di essere un facilitatore. Fa in modo che il team di progetto possa lavorare nelle migliori condizioni, senza ostacoli, focalizzato sugli obiettivi e sia a conoscenza delle metodologie utilizzate.

Eventi Scrum

Scrum prevede una serie di eventi organizzati secondo una *time box*, un periodo di tempo limite al quale ci si deve attenere e non si può sforare. L'obiettivo è quello di creare regolarità e minimizzare ulteriori incontri.

Il cuore della metodologia sono gli sprint. Lo sprint è un periodo di tempo di durata compresa tra due e quattro settimane durante il quale viene realizzato un incremento alle

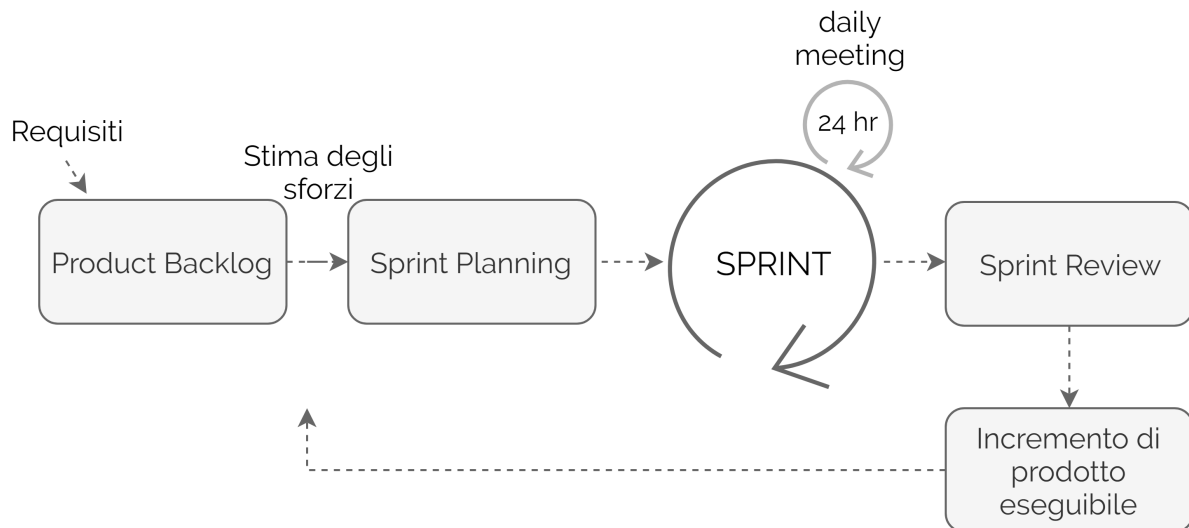


Figura 2.3: Rappresentazione del *framework* scrum

Fonte: adattato da Abrahamsson et al., 2003

funzionalità del prodotto. Al termine dello sprint, il prodotto potrebbe essere potenzialmente rilasciato. Vengono scelti degli obiettivi definiti da portare a termine durante la sua durata e ognuno di essi potrebbe quindi essere considerato come un progetto della durata massima di un mese.

Il *framework* scrum e gli elementi che lo compongono, rappresentati in Figura 2.8 vengono di seguito spiegati:

- **Product backlog:** è una lista ordinata di tutte le caratteristiche, funzioni, requisiti e miglioramenti necessari affinché il prodotto sia appropriato, utile e competitivo. Il *product backlog* è dinamico e evolve insieme al prodotto. Gli elementi vengono redatti dal *product owner* e dal team di sviluppo, definendo una priorità alle caratteristiche da sviluppare. Gli elementi con priorità maggiore saranno quelli svolti durante i primi sprint e saranno descritti con un grado di dettaglio maggiore rispetto ai successivi.
- **Sprint planning:** partendo dal *product backlog* e dalle performance degli sprint precedenti viene pianificato cosa verrà implementato durante lo sprint successivo. Il *product owner* definisce gli obiettivi che vorrebbe veder realizzati, che garantiranno un incremento di valore per il prodotto, mentre il team di sviluppo stabilisce il lavoro e lo sforzo necessari per realizzarli. Questo viene poi scomposto in attività della durata di circa una giornata.

- **Daily Scrum:** è un incontro della durata di quindici minuti che il team di sviluppo svolge ogni giorno. Serve a pianificare il lavoro della giornata successiva, ottimizzare la collaborazione del team, aumentare le performance valutando il lavoro realizzato dall'ultimo *daily scrum* ed evidenziare eventuali ostacoli.
- **Sprint Review:** viene svolta al termine dello sprint per verificare l'incremento di valore realizzato e adattare il *product backlog* quando necessario. Partecipa lo *scrum team* e gli *stakeholder* interessati. L'obiettivo è quello di ottenere feedback e favorire la collaborazione.
- **Sprint Retrospective:** viene svolta dal team e lo scopo è valutare lo sprint precedente, in particolare le persone, le relazioni, i processi e gli strumenti utilizzati, identificare gli elementi che sono andati bene e i potenziali miglioramenti e creare un piano per realizzarli durante lo sprint successivo. Costituisce un momento formale per focalizzarsi sull'autovalutazione e il miglioramento.

2.1.4.3 Lean Software Development

Lean è un *framework* derivante dal mondo produttivo che ha come obiettivo definire, costruire e consegnare un sistema software complesso che sia esattamente quello che il mercato chiede per restare competitivo. Come *Scrum*, non definisce delle vere e proprie pratiche utilizzabili nel mondo software e per questo si integra bene con altre metodologie.

Si basa su sette principi quali: eliminare gli sprechi, fornire valore, creare conoscenza, ritardare le decisioni all'ultimo momento possibile, fare consegne veloci, rispettare le persone e ottimizzare il tutto.

Una forte attenzione viene posta nella definizione dei giusti requisiti e del loro impatto nel business. In particolare il valore viene definito dal cliente e misurato dall'impatto su costi e ROI di progetto.

Altro elemento importante è dato dai membri del team, che devono avere le giuste capacità. Non solo devono essere a conoscenza degli aspetti sia funzionali che tecnici del sistema, ma devono anche collaborare con altri membri per comprendere a fondo il valore di business delle funzioni del sistema e i problemi che andranno a risolvere.

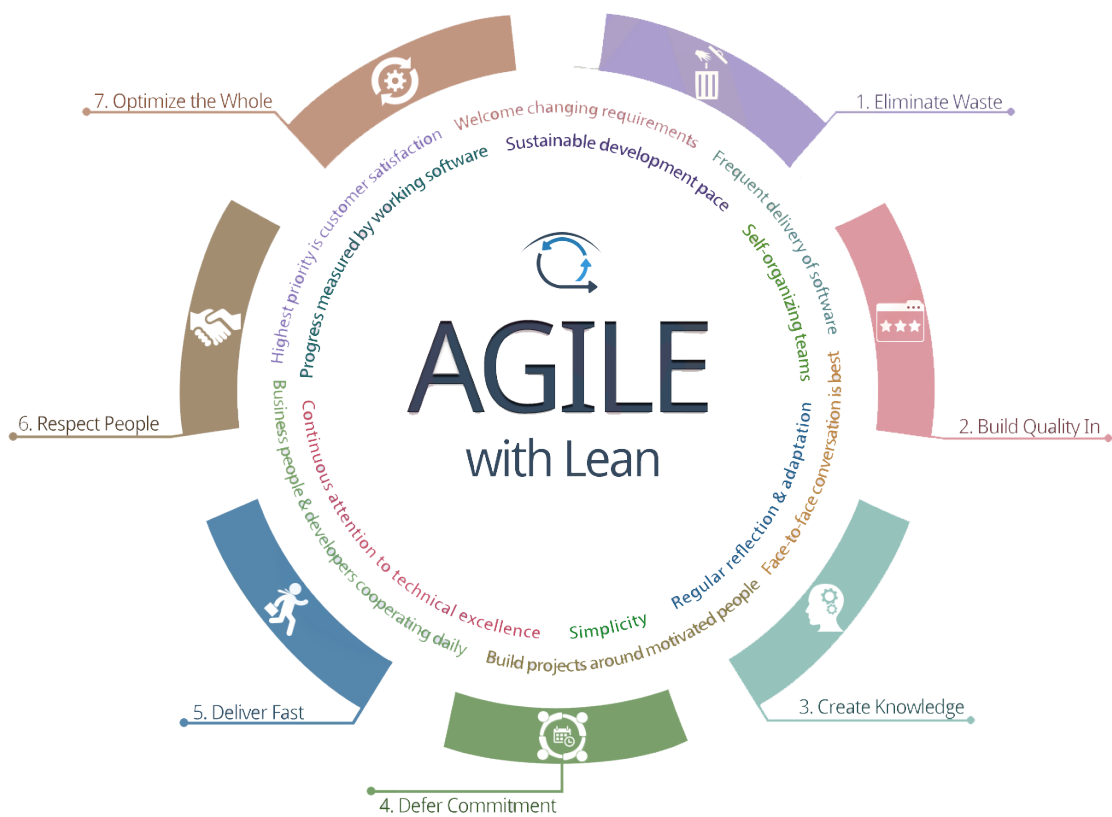


Figura 2.4: Valori *Lean Software Development*

Fonte: www.yodiz.com

2.2 L'approccio *agile* in altri settori

L'*agile project management* sta trovando sempre maggior interesse anche in industrie diverse da quella software. In rete è possibile trovare svariati esempi di aziende che utilizzano metodi agili per la gestione dei progetti, come dimostra anche il *Pulse of the Profession report* (2018) condotto dal PMI e illustrato in secondo il quale il 30% dei progetti in media utilizza un approccio *agile*, mentre il 23% una versione ibrida.

Percentuale media di progetti che usano approcci predittivi	44%
Percentuale media di progetti che usano approcci agile	30%
Percentuale media di progetti che usano approcci ibridi	23%
Percentuale di progetti che usano approcci "altro"	4%

Figura 2.5: Metodi usati per la gestione dei progetti nel 2018.

Fonte: adattato da *PMI's Pulse of the Profession report*, 2018

Tuttavia, mentre la letteratura per il mondo software è ormai estesa, non sono ancora presenti molti articoli accademici riguardo l'utilizzo, i benefici e le sfide delle metodologie in altri contesti. (Ciric et al., 2018)

L'attenzione sta crescendo e negli ultimi anni stanno cominciando a emergere vari studi su questo tema.

Gustavsson (2016) propone un'analisi della letteratura al fine di trovare esempi di applicazioni di tali pratiche e quali benefici e sfide siano stati riscontrati.

Un primo elemento interessante è rappresentato dalle pubblicazioni raccolte, che sono tutte successive al 2006, mentre il manifesto *agile* è stato diffuso dal 2001, a testimoniare come i primi interessi si siano manifestati ormai da alcuni anni.

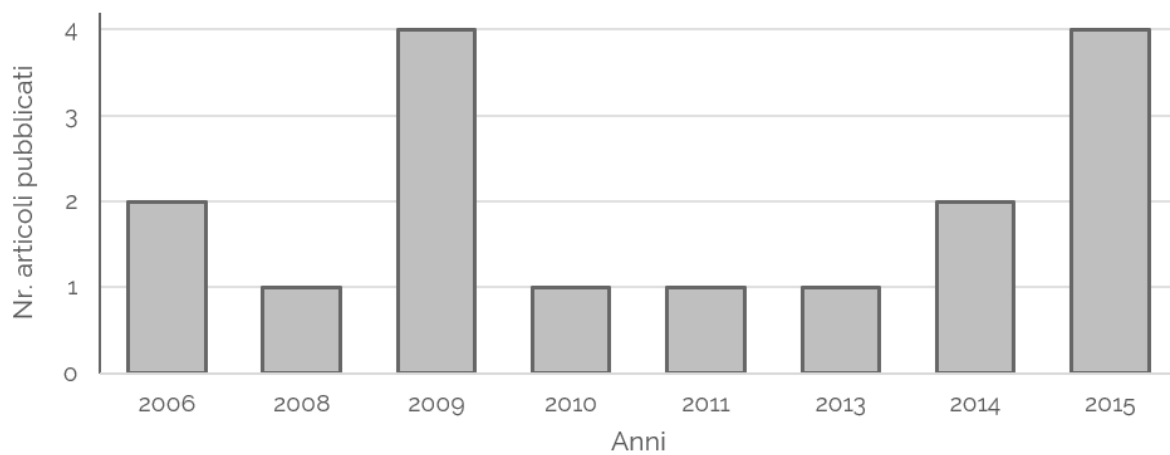


Figura 2.6: Numero di articoli pubblicati per anno.

Fonte: adattato da Gustavsson, 2016

I risultati emersi dai sedici articoli e i relativi ventuno casi analizzati per quanto riguarda benefici e sfide sono riportati in Figura 2.7.

#	Reported benefits	#	Reported challenges
11	Better collaboration in the team	3	Changing mindset to allow flexibility
9	Increased customer interaction	3	Lack of process visibility
8	Increased productivity and speed	2	Buy-in from managers
7	Increased flexibility , coping with change	2	Difficult to see benefits early in the project
6	Better understanding of goals/tasks/requirements	2	Inadequate knowledge sharing
6	Increased transparency and visibility	2	Individual work, lack of communication
5	Increased quality	2	Long-term planning
5	Customer-centered value-add priority process	1	Lack of stakeholder engagement
4	Increased knowledge sharing	1	Scope creep
3	Increased cross-organizational collaboration	1	Insufficient resource allocation
3	Better focus	1	Redundant work
2	Impediment removal process		
2	Increased individual autonomy		
2	Decreased customer complaints		
2	Increased motivation		
1	Clear sense of progress		
1	Improved resource allocation		

Figura 2.7: Benefici e sfide riportati negli articoli.

Fonte: Gustavsson, 2016

Nella maggior parte dei casi da lui analizzati i benefici riscontrati sono: una miglior collaborazione nel team, l'aumento del coinvolgimento del cliente, della produttività e della velocità. La maggior parte dei benefici si possono ricondurre al primo valore dell'*Agile Manifesto* "Individui e interazioni, più che processi e strumenti", mentre alcuni dei benefici emersi non sono direttamente riconducibili a valori o principi del Manifesto (come il miglior

focus nel progetto o il processo di rimozione degli ostacoli). Le sfide emerse dagli articoli sono invece poche, riguardanti principalmente il cambio di mentalità da effettuare e la mancanza di visibilità del processo.

Un altro spunto interessante viene offerto da Conforto et al. (2014) che studiano la possibilità di adottare un approccio *agile* in industrie diverse dal mondo software analizzando i fattori abilitanti all'uso di tale approccio e se sono presenti aziende che utilizzino alcune di queste pratiche, pur non essendo coscienti che possano essere ricondotte ad un'approccio *agile*. Il risultato ottenuto dimostra come alcune di queste pratiche siano già utilizzate all'interno delle imprese campione, come l'uso di strumenti per lo più *visual* per descrivere l'ambito di progetto, la creazione del piano di progetto in maniera collaborativa e con responsabilità condivisa e l'aggiornamento dello stesso con frequenza settimanale.

Altro elemento analizzato dalla letteratura è la possibilità di utilizzare un approccio ibrido *agile*-tradizionale per tipologie diverse di progetti. Riconoscendo la presenza di progetti e aziende con necessità e natura differente, una possibile soluzione potrebbe essere combinare benefici e difficoltà dei due approcci.

Sommer (2015) propone un'analisi di cinque aziende che hanno adottato un approccio ibrido, evidenziando come gli intervistati abbiano riscontrato un miglioramento qualitativo delle performance all'interno dell'azienda dopo l'adozione del sistema, riconducibile principalmente alla diffusione dei valori *agile*, agli effetti delle tecniche di comunicazione *visual*, alla condivisione della conoscenza e all'aumento della motivazione e produttività dei dipendenti.

2.3 Analisi delle implicazioni *agile* nella gestione dei progetti

2.3.1 Le aree di conoscenza nella gestione dei progetti

Un modo per analizzare la gestione dei progetti può essere quello di considerare i vari ambiti che lo coinvolgono. Ai fini di questa tesi è stato scelto di utilizzare le aree di conoscenza individuate dal *Project Management Institute*, che verranno di seguito riportate in maniera sintetica.

- **Gestione dell'integrazione di progetto**

Sono le attività necessarie a combinare e coordinare il progetto nel suo insieme. Alcune azioni da compiere sono la gestione delle interdipendenze tra le altre aree di conoscenza, il bilanciamento di esigenze contrastanti, l'esame di approcci alternativi e la capacità di adattare i processi per raggiungere gli obiettivi di progetto.

- **Gestione dell'ambito di progetto**

Comprende i processi utili alla definizione delle attività necessarie a raggiungere gli obiettivi del progetto con successo. L'intento è quello di individuare tutto e solo il lavoro richiesto dal progetto. Prevede la raccolta dei requisiti di progetto e dei bisogni degli *stakeholder*, la definizione dello scopo del progetto in maniera dettagliata, la creazione della *Work Breakdown Structure* e le operazioni di validazione e controllo.

- **Gestione dei tempi di progetto**

Include le attività necessarie a realizzare il completamento del progetto entro le tempistiche definite. È innanzitutto necessario definire le attività, ordinarle, individuando le relazioni tra di esse, stimarne la durata con le risorse assegnate, realizzare la pianificazione e monitorarne l'avanzamento.

- **Gestione dei costi di progetto**

Prevede la stima dei costi di progetto, la determinazione del budget e il loro controllo.

- **Gestione della qualità di progetto**

Comprende la pianificazione, la realizzazione e il controllo delle attività necessarie ad assicurare il raggiungimento del livello di qualità dei requisiti di progetto in modo da incontrare la politica aziendale e le richieste degli *portatori d'interesse*.

- **Gestione delle risorse di progetto**

Sono i processi volti a identificare, acquisire e gestire delle risorse, sia fisiche che umane, necessarie a completare il progetto con successo. L'obiettivo è assicurare al

project manager e al team di progetto le risorse necessarie al momento giusto e nel posto giusto. È necessario stimare la quantità della risorse e le attività a loro assegnate, acquisirle e controllarne l'effettiva disponibilità. Oltre a ciò, considerando le risorse umane, vanno previste attività per migliorarne le competenze e l'ambiente di lavoro al fine di favorire le performance di progetto, gestire il team e monitorare l'avanzamento dei compiti, ricevendo feedback e risolvendo i problemi.

- **Gestione delle comunicazioni di progetto**

Include le attività necessarie ad assicurare un efficace scambio di informazioni tra tutti i soggetti interessati e la realizzazione della documentazione richiesta.

- **Gestione dei rischi di progetto**

Gli obiettivi della gestione dei rischi sono di aumentare la probabilità di accadimento e/o l'impatto di rischi positivi (opportunità) e di ridurre quelli di rischi negativi (minacce), ottimizzando così le chance di successo del progetto. Le attività sono l'identificazione, l'analisi qualitativa e eventualmente quantitativa dei rischi e la pianificazione, l'implementazione e il monitoraggio di un piano di risposta agli stessi.

- **Gestione degli approvvigionamenti di progetto**

Comprende i processi utili ad acquistare o acquisire i prodotti e i servizi necessari da soggetti esterni. Include la gestione di contratti, ordini di acquisto e accordi sul livello di servizio.

- **Gestione degli *stakeholder***

Include le attività necessarie ad identificare le persone, i gruppi e le organizzazioni che possono essere impattati o impattare sul progetto, analizzando le loro aspettative e pianificando, gestendo e monitorando il loro coinvolgimento.

2.3.2 L'impatto *agile* nelle aree di conoscenza

I valori e i principi *agile* impattano su ognuna delle aree sopra descritte, influenzando la gestione stessa dei progetti. Di seguito verranno analizzati gli effetti nel dettaglio. L'analisi è tratta dall'*Agile Practice Guide* (PMI, 2017)

Gestione dell'integrazione di progetto

La gestione dell'integrazione è affidata ai membri del team e non al *project manager*; costoro hanno il compito di definire e decidere come i vari aspetti del progetto si integrano tra loro, gestendo eventuali conflitti tra le aree.

L'essenza del *project manager* non cambia in un ambiente flessibile, ma viene modificato il modus operandi con il quale ottiene i risultati di progetto. La definizione dell'ambito di progetto e la pianificazione dettagliata di tempi e compiti sono affidate al team, mentre il *project manager* si focalizza sul costruire un ambiente collaborativo e sull'assicurare che il team abbia l'abilità di rispondere in maniera efficace ai cambiamenti. Questo approccio collaborativo può venire rinforzato quando i membri del team posseggono delle conoscenze ampie anziché una stretta specializzazione, favorendo la visione complessiva del progetto e facilitando la collaborazione tra i membri.

Ciò si configura come una diretta applicazione dei principi *agile*:

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.

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

Gestione dell'ambito di progetto

In un contesto *agile* viene speso meno tempo per definire e accordare i requisiti di progetto nelle fasi iniziali, poiché in un ambiente con specifiche che evolvono, alto rischio o grande incertezza l'ambito di progetto potrebbe non essere chiaro fin dall'inizio o variare durante l'avanzamento. Molto spesso infatti ci si rende conto

della presenza di un *gap* tra i requisiti di prodotto desiderati effettivamente e quanto originariamente definito. Durante le fasi intermedie, di sviluppo e affinamento del progetto, viene però dedicato maggior tempo alla raccolta dei requisiti, alla definizione dell'ambito e alla creazione della WBS rispetto al processo tradizionale.

Il rilascio di versioni incrementali e la costruzione di prototipi contribuiscono alla definizione dell'ambito di progetto lungo la sua vita, rendendo possibile lo sviluppo degli elementi più chiari in un primo momento e posticipando la discussione delle funzionalità meno definite. Ciò viene reso ancor più efficace grazie al maggior contatto con il committente e/o il cliente, che hanno modo di verificare frequentemente che l'ambito di progetto sia in linea con le loro aspettative e contemporaneamente di rilasciare dei feedback sugli sviluppi futuri sulla base di quanto precedentemente realizzato.

I principi coinvolti sono:

La nostra massima priorità è soddisfare il cliente rilasciando fin da subito e in maniera continua software di valore.

Accogliamo i cambiamenti nei requisiti, anche durante 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.

Gestione dei tempi di progetto

Per gli stessi motivi descritti al punto precedente, anche la pianificazione dei tempi avviene in maniera incrementale.

Il lavoro è suddiviso e pianificato per brevi cicli di tempo, al termine dei quali vengono rivisti i risultati e realizzati gli adattamenti necessari. Le due principali modalità con cui ciò viene raggiunto sono:

- **Pianificazione iterativa** con un *backlog*:

I compiti vengono pianificati in maniera differente a seconda della fase del ciclo di vita del progetto: durante i primi incontri strategici, quando le informazioni

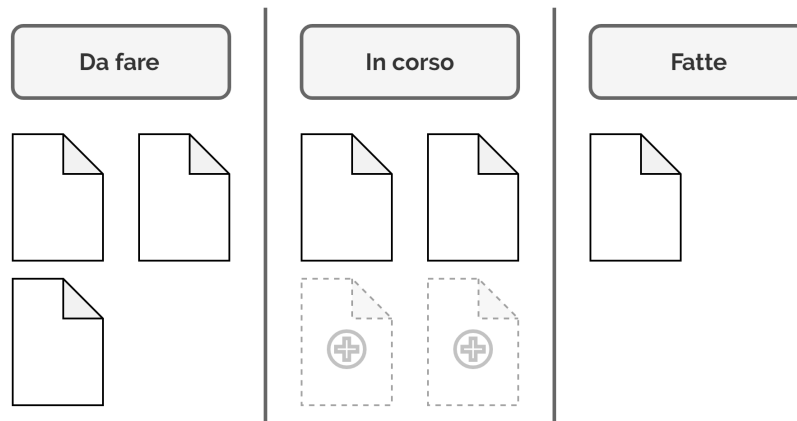
sono meno definite, i *work packages* vengono scomposti in maniera non troppo dettagliata, solo ad alto livello, concretizzandosi in storie utente e andando a formare il *backlog*. Mano a mano che si acquisiscono maggiori conoscenze, con il procedere del progetto, i *work packages* acquisiscono maggiore profondità e grado di dettaglio. In particolare viene assegnata una priorità alle storie utente; vengono definite quelle che verranno sviluppate nel periodo di tempo immediatamente successivo, che ha una durata fissa detta *time-box*, e vengono scomposte in attività più elementari.

L'approccio è spesso usato per consegnare valore incrementale al cliente o quando team multipli possono sviluppare contemporaneamente un grosso numero di caratteristiche che hanno poche dipendenze interconnesse.

- **Pianificazione su richiesta:**

Questo approccio, tipicamente usato nel sistema *Kanban*, è basato su una pianificazione con logica *pull* derivante dalla *lean manufacturing*. L'obiettivo è limitare i compiti in corso in modo da bilanciare le richieste di sviluppo con le attività che il team riesce a portare effettivamente a termine. Le mansioni non sono "tirate" da una pianificazione a monte e non hanno un intervallo di tempo predefinito durante il quale devono essere portate a termine, ma vengono scelte da un *backlog* e prese in carico non appena sono stati completati i compiti precedenti e, quindi, sono disponibili risorse.

Anche questo strumento di pianificazione viene spesso usato per progetti che si possono sviluppare in maniera incrementale, meglio se con compiti di dimensioni e scopo simile o raggruppabili per tali caratteristiche.

Figura 2.8: Esempio di una *kanban board*.

Fonte: propria elaborazione

Come nel punto precedente, i principi a cui ci si riferisce maggiormente sono:

Accogliamo i cambiamenti nei requisiti, anche durante 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.

Gestione dei costi di progetto

A causa dei frequenti cambiamenti, progetti con un alto grado di incertezza o con uno scopo non completamente definito fin dall'inizio potrebbero non beneficiare di una stima dettagliata dei costi fin dalle prime fasi. Risultano invece più indicati dei metodi di stima leggeri in un primo momento, per avere una previsione veloce e di alto livello che può venire aggiustata rapidamente quando il cambiamento avviene. Stime dettagliate sono invece riservate a una pianificazione di breve termine secondo un approccio *just-in-time*. Nel caso di progetti ad alta variabilità soggetti a budget severi, solitamente scopo e pianificazione dei tempi vengono aggiustati per rispettare i vincoli di costo.

In generale, i costi non rappresentano un punto di focus nella cultura *agile*, non venendo citati in nessuno dei principi.

Gestione della qualità di progetto

Al fine di accogliere il cambiamento, le metodologie agili verificano spesso la qualità e realizzano revisioni lungo tutto il progetto, anziché alla fine.

I controlli frequenti e l'interazione con il cliente garantiscono una verifica ripetuta della qualità del progetto, assicurando che rispecchi le aspettative del cliente e degli altri *stakeholder*. Retrospective frequenti controllano regolarmente l'efficacia dei processi. Durante il loro svolgimento viene cercata la causa radice dei problemi e vengono suggerite possibili soluzioni per migliorare la qualità. In quelle successive vengono verificati i processi cui era stata proposta una modifica, allo scopo di determinare se stanno funzionando e possono continuare, sono necessari aggiustamenti o se devono venire abbandonati. Per facilitare consegne frequenti e incrementali i processi agili si focalizzano su piccoli lotti di lavoro, scegliendo quelli che soddisfano maggiormente gli obiettivi di progetto. Ciò ha l'obiettivo di scoprire incongruenze e problemi di qualità nelle prime fasi della vita del progetto dove i costi dei cambiamenti sono più bassi.

La qualità di progetto è sicuramente un aspetto con un impatto rilevante nell'approccio *agile* e viene ripreso dai seguenti principi:

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

A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

Gestione delle risorse di progetto

L'approccio *agile* ha dato il via alla nascita di team auto-organizzati, dove il team lavora in assenza di un controllo centralizzato. In questi progetti, il *project manager* (che potrebbe essere chiamato in un altro modo) ha il compito di fornire al team l'ambiente e il supporto necessario, oltre che la fiducia, per portare a termine il lavoro. Per avere successo questi team sono interfunzionali e formati da specialisti che hanno sviluppato competenze trasversali, che continuano ad adattarsi ai cambiamenti dell'ambiente e abbracciano i feedback costruttivi.

La collaborazione è volta a migliorare la produttività e facilitare la soluzione di problemi in maniera innovativa. I team collaborativi possono facilitare l'integrazione di diverse attività, migliorare la comunicazione, aumentare la condivisione di conoscenza e fornire flessibilità all'assegnazione dei compiti di lavoro. Sebbene i benefici della collaborazione si applicano anche ad altri ambienti di lavoro, in questi progetti la capacità e la possibilità del team nel prendere decisioni sono critici al raggiungimento dei risultati prefissati, perché potrebbe non esserci il tempo per centralizzare i compiti e prendere decisioni.

Un team *agile* dovrebbe essere in grado di lavorare in maniera continuativa e costante con un ritmo stabile e prevedibile. L'obiettivo è facilitare la pianificazione dei compiti e non usare in maniera eccessiva le risorse per evitare di giungere a situazioni di logoramento psicofisico dovuto alla mancanza di energie e di capacità per sostenere e scaricare lo stress accumulato. Sono consigliate le quaranta ore settimanali di lavoro.

Pianificare risorse fisiche e umane è molto meno prevedibile in progetti ad alta variabilità. In questi ambienti gli accordi di approvvigionamento veloce e i metodi *lean* sono critici per controllare i costi e rispettare la pianificazione.

Questi elementi trovano riscontro nei seguenti principi:

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.

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

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

Gestione delle comunicazioni di progetto

In ambienti soggetti ad ambiguità e cambiamenti l'importanza della comunicazione è ancora maggiore vista la necessità di dover trasmettere i dettagli in evoluzione e

in cambiamento il più frequentemente e velocemente possibile. Per questo i membri del team devono avere accesso alle informazioni, fare incontri frequenti e trovarsi nello stesso ambiente fisico il più possibile.

Gli incontri per mostrare i risultati del progetto sono attività chiave per promuovere la trasparenza con il management e gli *stakeholder*.

Il concetto viene esplicitato dal principio:

Il modo più efficiente e più efficace per comunicare con il team e all'interno del team è la conversazione faccia a faccia.

Gestione dei rischi di progetto

La gestione dei rischi non è un'aspetto considerato direttamente dai principi *agile*, sebbene ambienti ad alta variabilità, per definizione, siano maggiormente esposti ad incertezza e rischio. Le revisioni frequenti dei prodotti realizzati in maniera incrementale tuttavia, accelerano la condivisione della conoscenza e aiutano a mitigare, comprendere e gestire il rischio. La pianificazione per iterazioni fa sì che il rischio venga considerato quando si selezionano i compiti da svolgere nel periodo di tempo successivo. I rischi vengono cioè identificati, analizzati e gestiti durante ogni iterazione.

Inoltre, l'aggiornamento regolare dei requisiti e la ridefinizione delle priorità dei compiti sono attività che possono venire svolte più volte durante la vita del progetto, adattandosi alla migliore comprensione dell'esposizione al rischio attuale.

Gestione degli approvvigionamenti di progetto

In ambienti *agile* specifici fornitori potrebbero essere usati come estensione del team. Questa relazione collaborativa può portare a una condivisione dei rischi e delle ricompense connessi al progetto.

Progetti complessi e di grandi dimensioni potrebbero usare un approccio ibrido per la gestione dei contratti di fornitura, usando logiche tradizionali per la parte più stabile e riservando la parte variabile a un'appendice o un supplemento contrattuale. In questo modo cambiamenti di scopo possono avere luogo senza impattare sul contratto complessivo.

Gestione degli *stakeholder*

I progetti con un alto grado di cambiamento richiedono il coinvolgimento e la partecipazione attiva degli *stakeholder*. Per facilitare discussioni e decisioni tempestive e produttive, i team flessibili si relazionano direttamente con i soggetti interessati piuttosto che rivolgersi a diversi livelli di *management*. Spesso il cliente, l'utente e lo sviluppatore si scambiano informazioni in un processo dinamico e di co-creazione che porta a un maggior coinvolgimento degli *stakeholder* e una loro maggiore soddisfazione. Le interazioni regolari lungo tutto il progetto mitigano il rischio, creano fiducia e consentono cambiamenti fin dalle prime fasi del ciclo di progetto, riducendo i costi di modifiche che potrebbero avvenire in fasi avanzate e incrementando le probabilità di successo del progetto.

Al fine di accelerare la condivisione delle informazioni all'interno e all'esterno dell'azienda, le metodologie *agile* promuovono la trasparenza tramite la presenza degli *stakeholder* ai meeting di progetto e alle *review* oppure la condivisione della documentazione di progetto in luoghi accessibili a tutti gli interessati in modo da far emergere il più presto possibile eventuali disallineamenti, dipendenze o altri problemi collegati ai cambiamenti.

I principi che riprendono il tema sono:

Accogliamo i cambiamenti nei requisiti, anche durante stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.

Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.

Il modo più efficiente e più efficace per comunicare con il team e all'interno del team è la conversazione faccia a faccia.

3 | Il caso Galdi

3.1 L'azienda

Galdi è un'azienda produttrice di macchine automatiche per il confezionamento di liquidi in cartoncini *gable top*⁵ che vende a livello globale. È una media impresa, conta un totale di 105 dipendenti, di cui 89 nella sede italiana, con un fatturato che nel 2018 si aggirava intorno ai 20 milioni di euro.

La forza di Galdi è l'elevata flessibilità delle sue macchine riempitrici, si differenzia infatti dalla concorrenza offrendo un'elevata varietà di prodotti dedicati a tirature medio/piccole, dall'elevata personalizzazione, in grado di confezionare un'enorme gamma di formati di cartone e riempire diversi prodotti, che variano da latte e prodotti dairy, freschi o ESL (*Extended Shelf Life*), passando per le uova liquide fino a bevande come succhi di frutta o vino.

3.1.1 La storia

L'azienda nasce dall'idea di Galdino Candiotta che, per aumentare la produttività della latteria di famiglia, decide di progettare da sé una macchina riempitrice semiautomatica, successivamente utilizzata anche da altre latterie della zona.

Nel 1985 fonda l'azienda Galdi, specializzata nella realizzazione di macchine automatiche. Da quell'anno l'azienda continua ad operare, crescere e evolversi.

Nel 1993 viene venduta la prima macchina al di fuori del mercato nazionale.

In concomitanza con il cambio di direzione dell'azienda, che trasferisce la gestione alla nuova generazione, nel 2010 Galdi intraprende il percorso *lean* in ambito produttivo, che le consentirà di ottimizzare spazi e risorse, riducendo gli sprechi e raggiungendo ottimi risultati produttivi e qualitativi.

Da qualche anno a questa parte l'azienda si è invece approcciata all'*agile* in ambito sviluppo nuovi prodotti, per cercare di mantenere il passo e reagire al cambiamento promuovendo nuovi risultati.

⁵letteralmente "tetto a timpano". Fa riferimento alla caratteristica forma triangolare del tetto dei cartoncini, simile a una casetta, che si differenzia da quelli con tetto piatto.

Vision di Galdi è:

"Miriamo a semplificare la vita dei clienti progettando servizi e soluzioni di confezionamento concepiti attorno alle loro esigenze."

Negli ultimi anni infatti è cresciuta sempre più l'importanza di offrire un servizio completo al cliente, trasformandosi da semplici fornitori a un'azienda in grado di accompagnare il cliente dalla scelta della riempitrice ottimale per soddisfare le sue esigenze alla copertura del servizio post-vendita.

Per questo motivo nell'ultimo decennio l'azienda ha deciso di fondare diverse *business unit* strategiche nei maggiori mercati di riferimento in modo da avere un contatto più vicino con il cliente: nel 2011 in Russia, nel 2013 in Marocco, nel 2015 negli USA e nel 2018 in Algeria.

3.1.2 Il prodotto

Come si accennava nell'introduzione al capitolo, Galdi è un'azienda produttrice di macchine automatiche per il confezionamento di liquidi in cartoncini *gable top*, specializzata in tirature medio/basse e con un prodotto altamente personalizzabile.

Il prodotto dell'azienda si va a definire tramite la combinazione delle seguenti caratteristiche principali, a cui si associano poi una serie di optional:

- Capacità produttiva

L'azienda produce tre serie principali (RG270, RG50, RG21) a cui corrisponde una capacità produttiva decrescente (rispettivamente fino a 7000, 3000, 2000 pezzi/ora)

- Formato del cartone

Le sezioni di cartone più comuni sono mini (57x57 cm), standard (70x70 cm), e due litri (70x95 cm, 91x91 cm, 95x95 cm). Alla sezione corrisponderà poi una certa altezza del cartoncino, che andrà così a determinare il volume riempibile.

- Tipo di prodotto e *shelf life* desiderata:

A seconda di queste caratteristiche viene scelta la tipologia del sistema di riempimento e del sistema di sanificazione della macchina.

A ciò si vanno ad aggiungere una serie di elementi di optional, quali potrebbero essere applicatore tappi o datario, che vanno ad espandere ulteriormente la varietà dello spazio prodotto. Questo può dare un'idea del contesto di estrema varietà in cui l'azienda si trova ad operare e delle numerose opzioni configurabili che si vanno a delineare una volta definita la serie macchina.

Oltre a ciò, Galdi realizza anche macchine per alcuni partner, che vengono vendute con il loro marchio.

3.1.3 Il principio di funzionamento del prodotto realizzato

Il principio di funzionamento è comune a tutte le serie di macchine e consente di ottenere a partire dal cartone preformato appiattito, il cartone formato, chiuso, con il tappo applicato quando previsto e riempito del prodotto desiderato. Una volta impostati i parametri di input, il processo avviene in maniera completamente automatica.

- Prelievo e carico:

Il primo cartone viene prelevato dal magazzino tramite un sistema a vuoto, viene formato e caricato nel primo mandrino della giostra tramite un sistema meccanico.

- Formazione del fondo:

Durante la formazione del fondo i cartoni sono supportati dai mandrini per mantenerne la forma e spostati nelle varie stazioni dalla giostra. Il fondo del cartone viene prima riscaldato tramite aria calda, in modo da sciogliere il sottile strato di polietilene interno al cartone. Successivamente avvengono la prima piega e la pressatura del fondo, che vanno a sigillare la parte inferiore del cartone sfruttando il raffreddamento della plastica leggermente sciolta.

- Scarico del cartone:

Il cartone viene trasferito dalla giostra e posizionato fra le alette della catena di trasporto, che ne mantengono la forma e lo guidano attraverso le fasi successive.

- Applicazione del tappo:

Se previsto, il tappo viene prelevato dall'apposito magazzino, posto in prossimità della macchina, orientato e spinto all'interno del cartone dove viene saldato tramite un sistema ad ultrasuoni.

- Disinfezione:

Se previsto, il cartone viene disinfettato in modo tale da eliminare eventuali batteri presenti. L'obiettivo è quello di preparare il cartone al riempimento in modo da garantire che il processo all'interno della macchina non vada a deteriorare le condizioni del prodotto da riempire.

- Dosatura:

Il cartone viene riempito con il prodotto desiderato per la quantità corretta.

- Chiusura del tetto:

La parte superiore del cartone viene piegata, riscaldata e saldata tramite un principio simile alla formazione del fondo, facendo maggior attenzione a non contaminare il prodotto.

- Uscita del cartone:

Il cartone esce dalla macchina e viene posizionato sul breve tratto di nastro d'uscita collegato alla linea produttiva del cliente.

3.1.4 Il processo produttivo



Figura 3.1: Schema processo produttivo delle macchine

Fonte: propria elaborazione

Il processo produttivo delle macchine Galdi prevede le seguenti macro-fasi, riportate schematicamente in Figura 3.1:

- Generazione dell'ordine interno

Una volta ricevuto l'ordine cliente, viene generato l'ordine interno. Ad ogni commessa viene assegnato un responsabile di progetto, che ha il compito di verificare la gestione dei tempi e interfacciarsi con il cliente, ad esempio, richiedendo i campioni per i test, assicurandosi che il cantiere dal cliente abbia tutte le *utilities* necessarie e organizzando le visite in azienda.

- Progettazione

In questa fase avviene la progettazione meccanica, elettrica e software delle macchine. L'obiettivo è quello di creare le distinte base e i disegni di assieme che serviranno alle fasi successive.

- Approvvigionamento materiali

Vengono emessi gli ordini di acquisto e di produzione interna.

Gli ordini di acquisto sono gestiti in maniera separata distinguendo tra quelli per articoli a commercio e quelli a disegno, che richiedono maggior comunicazione con il fornitore per la richiesta di fattibilità dei pezzi e l'invio dei disegni dei pezzi da realizzare.

Gli ordini di produzione interna invece sono relativi ai codici realizzati dal reparto carpenteria. È richiesta la pianificazione del lavoro e delle attività. Oltre ai componenti durante questa fase avviene anche la produzione del telaio. Una volta trascorso il tempo d'attesa necessario i materiali vengono ricevuti, accettati e caricati a magazzino.

- Assemblaggio

L'assemblaggio delle macchine avviene in linea su postazione fissa. Il telaio viene posizionato in linea durante il primo giorno di assemblaggio, mentre i materiali necessari ad assemblare le stazioni immediatamente successive del ciclo di lavoro sono prelevati da magazzino e posti su dei carrelli in appositi spazi a inizio linea secondo una sequenza definita.

Sono attive contemporaneamente fino a tre linee, servite tutte da una baia di pre-assemblaggio comune.

- Collaudo

Viene verificato il funzionamento della macchina, vengono apportate le tarature dovute e sono eseguiti i test necessari a verificare la qualità del cartone (forma corretta, tenuta delle saldature). Il collaudo è dotato di 6 baie, una delle quali viene utilizzata dal *service* per ricondizionare macchine usate.

- *FAT - Final Acceptance Test*

Il cliente viene invitato in azienda a verificare il funzionamento della macchina.

- Pulizia, Imballaggio e Spedizione

Infine, la macchina viene pulita, imballata e preparata alla spedizione, che potrebbe essere verso il cliente finale o verso un punto intermedio.

3.2 La gestione dei progetti di sviluppo nuovi prodotti in Galdi

Negli ultimi anni Galdi è passata da uno sviluppo prodotti in ottica *design to order* all'*assembly to order*, proponendo delle configurazioni standard, dette *bundle*, tra le quali il cliente può scegliere.

Ciò è stato possibile grazie alla modularizzazione dei componenti e alla realizzazione di progetti piattaforma (ad esempio per la serie RG270). Lo scopo è quello di ridurre il *lead time* di consegna, mantenere l'elevata varietà offerta ai clienti, semplificando gestione e sviluppo interno.

Buona parte degli ordini dei clienti sono quindi riconducibili a macchine *bundle*, non richiedono nuovi sviluppi e seguono il normale processo produttivo, che prevede tempi standard per le varie fasi, differenziati a seconda della serie e di alcune caratteristiche macchina.

L'azienda riconosce però la necessità di continuare a innovare e investire, per questo, alle commesse dedicate alle macchine *bundle* si affianca lo sviluppo di nuovi prodotti.

Già da alcuni anni Galdi ha intrapreso un cammino con alcuni elementi ispirati all'*agile* per la realizzazione di nuovi progetti, in particolare ha implementato un metodo personalizzato, chiamato in azienda *Lean Agile*, che verrà descritto nei paragrafi seguenti.

Forti dei principi base dell'*agile*, l'azienda ha cercato di far proprie alcune metodologie e applicarle alla realtà in cui opera, chiaramente differente dallo sviluppo software,

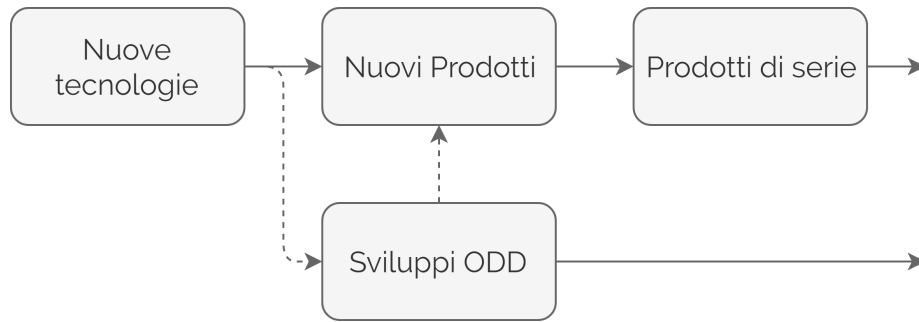


Figura 3.2: Tipologie di progetti

Fonte: documentazione interna all'azienda

affrontando e cercando di superare le varie criticità incontrate.

3.2.1 Classificazione progetti di sviluppo nuovi prodotti

Per quanto riguarda i progetti di sviluppo nuovi prodotti, questi sono stati classificati in tre macrocategorie:

- Sviluppo di nuove tecnologie (*New Technology Development* – NTD): progetti di sviluppo dedicati alla realizzazione di nuove tecnologie da impiegare in futuro nei nuovi prodotti.
- Sviluppo di nuovi prodotti (*New Product Development* – NPD): progetti di sviluppo di piattaforme o di prodotti singoli che verranno successivamente inseriti a catalogo.
- Sviluppi su richiesta (*On Demand Development* – ODD): progetti di sviluppo nuovi prodotti realizzati su richiesta di un particolare cliente. Possono essere combinazione di tecnologie e gruppi funzionali già esistenti o richiedere alcuni sviluppi. Sono prodotti estremamente personalizzati, tali da non giustificare la realizzazione di azioni volte a standardizzare. All'interno di questa categoria vengono fatti ricadere anche i progetti che prevedono una particolare declinazione di una piattaforma.

In Figura 3.2 vengono illustrate le possibili strade di sviluppo che può seguire un progetto. I progetti di sviluppo di nuove tecnologie servono come abilitatori per portare innovazioni tecniche negli altri progetti. Lo sviluppo di nuovi prodotti è finalizzato a creare nuove macchine da inserire a portafoglio. Un discorso particolare va fatto per i progetti ODD, che seguono un loro percorso *stand alone*, ma potrebbero includere al loro interno sviluppi di nuove tecnologie e potrebbero anche fornire la base per futuri sviluppi di nuovi prodotti.

3.2.2 Nascita di un progetto

La nascita di un progetto dalla concezione dell'idea al suo avvio segue alcune fasi, sintetizzate in Figura 3.3:

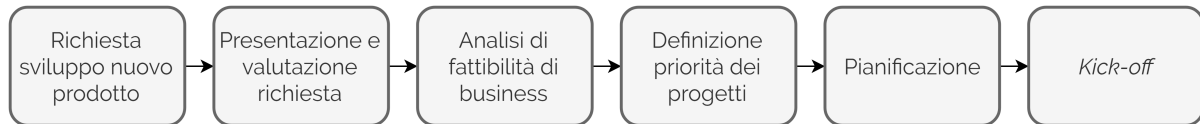


Figura 3.3: Fasi di avvio di un progetto

Fonte: documentazione interna all'azienda

Richiesta sviluppo nuovo prodotto

L'idea di un progetto può nascere da varie fonti sia interne che esterne. In particolare può partire da richieste di clienti, idee di miglioramento o di innovazione o la necessità di adattarsi a particolari normative esistenti (essendo macchine per il confezionamento di prodotti alimentari le normative sono particolarmente stringenti e articolate in maniera differente per ogni paese).

L'idea deve poi concretizzarsi nella richiesta di sviluppo di un nuovo prodotto, che viene diffusa in azienda tramite un format standard.

Presentazione e valutazione richiesta

Le richieste di nuovi prodotti e le opportunità commerciali vengono presentate e valutate dal *Product Development Team* composto dai responsabili dell'area tecnica e dell'area marketing.

L'obiettivo è quello di rendere tutti gli interessati a conoscenza delle informazioni e di effettuare una prima scrematura dei possibili progetti, in modo da non impiegare sforzi e risorse in progetti non rilevanti o in opportunità non concretizzabili.

Analisi di fattibilità di business

Durante l'analisi di fattibilità viene valutata la complessità tecnica dell'idea proposta, viene determinato il bin di appartenenza e quindi una durata indicativa del progetto e viene svolta una prima analisi dei costi.

Il concetto di *Bin* di appartenenza merita un approfondimento. Il Bin rappresenta un indice della complessità del progetto, che dipende dal numero e dall'entità delle modifiche apportate ai diversi gruppi funzionali, a cui è associato un tempo di sviluppo indicativo. Progetti più semplici (con meno modifiche e di entità minore) hanno tempi di sviluppo più brevi, progetti più complessi hanno tempistiche più lunghe.

La definizione del Bin di appartenenza del progetto viene determinata a partire dai Bin assegnati ai gruppi funzionali.

Il Bin di ogni gruppo funzionale viene individuato attraverso un complesso sistema di matrici a cascata. L'obiettivo è quello di considerare l'impatto delle variabili che influenzano la complessità di sviluppo del gruppo. Per ogni gruppo funzionale viene assegnato un primo livello di complessità sulla base delle modifiche ai parametri tecnici prestazionali, successivamente viene individuato un ulteriore livello di complessità associando anche la complessità tecnica dello sviluppo, ovvero il grado di novità tecnologica o di novità del concept. Infine viene considerata la complessità progettuale, ovvero il grado di novità tecnologica per l'azienda.

Sulla base del numero e dell'entità delle modifiche apportate ai gruppi funzionali viene quindi definito il Bin di appartenenza del progetto.

Come si può vedere in Tabella 2, la complessità dell'intero progetto viene assegnata sulla base dei gruppi funzionali più critici.

Ad esempio, un progetto con qualche modifica leggera (di grado A) ad un numero di gruppi funzionali minore di sette, verrà classificato come un progetto S1. Se è presente una modifica più importante ad un gruppo funzionale (grado C), il progetto verrà classificato in un bin S2, e così via.

Questo metodo di definizione dei bin, sebbene molto preciso, risulta essere un po' macchinoso da eseguire, specialmente nella parte di definizione della complessità delle modifiche ai gruppi funzionali, e per questo, molto spesso, il bin di appartenenza viene assegnato per confronto con altri progetti simili.

Ai Bin è stata poi assegnata una durata indicativa del progetto, indicata come una maggiorazione rispetto ai tempi standard, vedi Tabella 3.

Bin di progetto		Modifiche massime	Complessità dei gruppi funzionali				
			A	B	C	D	E
Small	S1	≤ 7	≤ 2	no	no	no	
	S2	≤ 11		≤ 1	no	no	
	S3	≤ 14		$2 \leq 3$	no	no	
Medium	M1	≤ 14		$4 \leq 5$	no	no	
	N2	≤ 14			≤ 2	no	
Large	L	≤ 14			≤ 5	≤ 2	

Tabella 2: Definizione del Bin di progetto

Fonte: documentazione interna all'azienda

BIN di progetto	Tipologia	Lead Time
S1	ODD	LT std + 2 settimane
S2	ODD	LT std + 4 settimane
S3	ODD	LT std + 8 settimane
M1	ODD	LT std + 16 settimane
M2	NTD o NPD	LT std + 32 settimane
L	NTD o NPD	LT std + ?? settimane

Tabella 3: Tempi associati ai Bin

Fonte: documentazione interna all'azienda

Questo consente di avere una durata indicativa e fin da un primo momento dei tempi di progettazione complessivi, che verrà poi dettagliata maggiormente in seguito. Per i progetti classificati con un bin L non è possibile individuare una durata relativamente precisa, in quanto sono presenti alcuni gruppi funzionali di cui non si conosce il lavoro da svolgere. È necessario perciò uno studio più approfondito di quei gruppi e solo in seguito sarà possibile definire delle tempistiche indicative.

Si è voluto anche imporre delle limitazioni ai progetti ODD, che non possono essere maggiori di un progetto M1, poiché avrebbero altrimenti tempi di sviluppo e impiego di risorse troppo elevati, riducendone anche il rischio. All'aumentare del numero e dell'entità delle modifiche da apportare ai gruppi funzionali aumenta infatti anche il rischio legato al progetto.

Valutazione analisi e definizione priorità

I risultati dell'analisi vengono nuovamente presentati al *Product Development Team*.

A questo punto viene definita una priorità alle idee in base a:

- attinenza alla strategia aziendale
- potenziale per la crescita dell'azienda
- competenze e tecnologie già disponibili o da sviluppare
- urgenza della richiesta

Viene quindi stabilito quali idee daranno il via ad un progetto di sviluppo nuovo prodotto e quali invece verranno scartate.

Pianificazione

Viene infine assegnato un *Project Leader* e determinato il team di progetto, effettuata una pianificazione con maggior grado di dettaglio, definendo delle *milestone* per le macrofasi (denominate IP) e inserito il progetto nel portafoglio di sviluppo nuovi prodotti.

Kick-off

A questo punto il progetto viene presentato all'azienda e ne viene dato il via ufficiale.

3.2.3 Esecuzione del progetto

Per ogni gruppo funzionale vengono svolte alcune macro-fasi illustrate in Figura 3.4. Queste fasi si rifanno al classico modello Stage-Gate per lo sviluppo dei prodotti, ma possono venire svolte in maniera iterativa, quando lo sviluppo lo richiede in un'ottica agile.

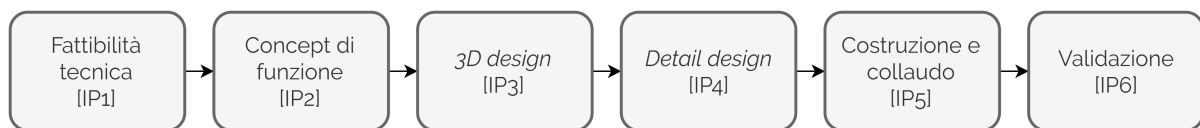


Figura 3.4: Fasi della vita di un progetto

Fonte: documentazione interna all'azienda

Fattibilità tecnica

Viene esplorata la fattibilità tecnica di diverse alternative, in maniera approssimativa con veloci calcoli. È importante esplorare un gran numero di idee, in modo tale da non precludere nessuna strada.

Concept di funzione

Vengono realizzate in maniera concreta le idee migliori generate nella fase precedente.

In questa fase è auspicabile l'utilizzo di test al fine di valutare l'effettivo funzionamento di quanto pensato.

È importante documentare le attività svolte, in modo da generare conoscenza che sia utilizzabile in futuro.

3D design

Viene realizzato il modello 3D del concept scelto nella fase precedente.

Detail Design

Vengono realizzate le tavole 2D di tutti i componenti. In questa fase vengono verificate le tolleranze dei pezzi e il risultato finale è la generazione delle distinte.

Costruzione e collaudo

In seguito alla progettazione, il prototipo si ricongiunge con il normale processo produttivo, andando ad utilizzare le stesse risorse destinate alla normale produzione. I progettisti svolgono una funzione di supporto all'assemblaggio e al collaudo, apportando eventuali modifiche ai componenti e collaborando per risolvere le varie problematiche che si vengono a creare.

Validazione

Infine, viene testato il reale funzionamento della macchina sul campo da un cliente, che può essere stato individuato a monte dell'inizio del progetto o cercato durante il suo svolgimento. La ricerca di un cliente adatto è un elemento importante, in quanto deve essere disposto a fornire dati e feedback sul funzionamento della macchina, prediligendo comunque clienti localizzati geograficamente in prossimità di Galdi, in modo da favorire eventuali ispezioni o insorgenza di problemi da risolvere.

Alle fasi di progettazione si è tentato di implementare lo sviluppo incrementale e iterativo tipico dell'agile. L'obiettivo era quello di ridurre eventuali errori, individuandoli fin dalle prime fasi di progettazione evitando di rendersene conto solo una volta arrivati al collaudo della macchina, aumentando contemporaneamente la qualità complessiva dei progetti.

Uno dei problemi è ovviamente quello di non poter realizzare e testare il prodotto finito in maniera incrementale, visti i lunghi tempi non solo di progettazione ma anche di approvvigionamento e realizzazione della macchina.

Per ovviare al problema si è proceduto con l'implementazione di cicli di test (Figura 3.5) ripetuti all'interno del processo di sviluppo. Questi test possono venire realizzati sia con tecniche di simulazione al computer, sia tramite l'utilizzo di banchi test appositamente creati, in maniera tale da consentire di valutare la bontà delle soluzioni ideate e apportare le opportune modifiche dopo aver valutato i risultati.

Queste fasi, sebbene originariamente prevedessero il coinvolgimento di altre figure, sono di fatto fortemente incentrate sul lavoro di progettazione meccanica.

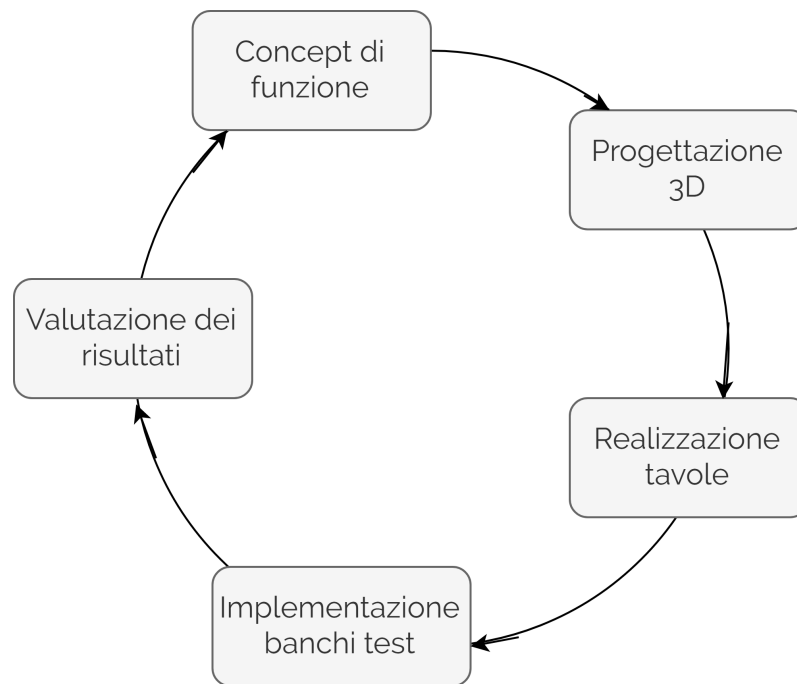


Figura 3.5: Esempio ciclo di test

Fonte: propria elaborazione

Al momento le altre figure, quali progettista elettrico, progettista software e automazione, responsabile della certificazione prodotto, responsabile della manualistica vengono coinvolte secondo la sensibilità del team leader, talvolta nel momento in cui la macchina sta per venire assemblata, con poco tempo a disposizione per lavorare al progetto.

3.2.4 Elementi di forza e di debolezza del processo

Dopo un paio d'anni di applicazione della metodologia e a fronte di un progetto complesso che ha presentato numerose difficoltà nell'attuazione, l'azienda ha deciso di analizzare nuovamente il processo introdotto in modo da individuare elementi positivi e elementi migliorabili.

Gli elementi di forza riscontrati sono:

- Cadenza dei progetti e organizzazione delle milestone (IP)

La presenza di milestone definite per le varie macro-fasi ha dato leggibilità ai progetti, consentendo di individuare rapidamente in che punto dello stadio di sviluppo ci si trovasse. L'assenza di una pianificazione dettagliata di breve periodo ha reso però difficile riuscire a raggiungere questi obiettivi.

- Realizzazione di simulazioni e test tramite banchi prova

Elemento apprezzato specialmente dai progettisti ha consentito di testare in anticipo e esplorare idee molto incerte, che non si sarebbero altrimenti proposte in macchina perché sconosciute e per questo rischiose. In assenza di questi banchi molte idee sarebbero rimaste inesplorate oppure testate solo in una fase molto avanzata comportando onerosi costi di redesign o soluzioni improvvisate realizzate per tamponare il problema in fase di collaudo.

I punti di debolezza principali sono invece:

- Cambio frequente dello scopo di fornitura del prodotto

La definizione approssimativa dei requisiti di prodotto combinata con l'assenza di un vero e proprio sistema per gestire le modifiche ai requisiti ha portato a variazioni continue dei requisiti non gestite, ma subite durante i progetti. Questo comportava insoddisfazione nei progettisti e difficoltà a rimanere motivati nei progetti.

- Scarsa gestione della pianificazione di dettaglio e di breve periodo

Se come si accennava tra i punti precedenti le milestone sono state riconosciute come un elemento positivo, l'assenza di un'accurata pianificazione di dettaglio e di breve periodo ha reso difficile concentrarsi sugli obiettivi immediati. Le milestone vengono poste come obiettivi intervallati a mesi di distanza, difficili da raggiungere e su cui è difficile trovare la motivazione per lavorare.

Questo, in aggiunta al continuo rinvio delle date degli obiettivi perché non venivano raggiunti, ha reso il clima poco sereno.

4 | L'adozione di Scrum e il progetto applicativo

4.1 Il progetto di miglioramento: obiettivi e strumenti

Gli obiettivi del progetto di miglioramento aziendale erano ottenere una maggior definizione dei requisiti di prodotto, ottenere la capacità di gestire eventuali modifiche ai requisiti e migliorare la gestione della pianificazione di dettaglio del progetto.

Per superare le criticità sopra elencate si è pensato di adottare in maniera più importante alcune delle metodologie *agile* e in particolare:

- utilizzo di *product canvas* per realizzare il concept di prodotto;
- utilizzo della metodologia *scrum* per la pianificazione di dettaglio;

Si è deciso di implementare i nuovi approcci in due progetti pilota, l'esperienza di uno dei quali è riportata in seguito.

4.1.1 Il *Product Canvas*

Il *product canvas* è uno strumento utile al fine di definire in maniera semplice, condivisa e intuitiva quali sono gli obiettivi e i requisiti del progetto.

Lo strumento nasce come una rielaborazione del *Business Model Canvas*, un *template* utile per definire in maniera visuale gli elementi più importanti di un nuovo modello di business.

Questo concetto, applicato alla gestione dei progetti, si concretizza nel *Product Canvas*, riportato in Figura 4.1.



Figura 4.1: Template del Product Canvas

Fonte: tratto da www.romanpichler.com

Le sezioni che lo compongono sono:

- Nome: il nome del progetto
- Obiettivo: descrizione della visione e dello scopo de progetto. Dovrebbe essere riassunto in un'unica frase.
- Metriche: elementi necessari a determinare il raggiungimento dell'obiettivo di progetto.
- Clienti target: in un'ottica *agile* ispirata a *XP* i cliente target possono venire descritti tramite metafore come *Personas*, ovvero caricature di tipologie di clienti, di cui viene descritto il *background* e le loro aspettative riguardo il prodotto/servizio.
- *Big Picture*: in questa sezione vengono descritti gli elementi e le caratteristiche fondamentali del prodotto. In ambiente *agile* e in particolare *XP* possono corrispondere alle epiche, ovvero alle macro storie che andranno sviluppate nel prodotto finito. Le storie utente sono una rappresentazione descrittiva della funzionalità che andranno implementate nel prodotto. Quando il prodotto è complesso più storie possono venire raggruppate tra loro, prendendo il nome di epica. L'obiettivo di utilizzare questo sistema è consentire una maggior leggibilità e comprensione anche a persone con *background* diverso da quello tecnico e mantenere il focus incentrato sul valore

che viene offerto al cliente piuttosto che sulle caratteristiche tecniche. Ciò viene facilitato dall'uso di frasi descrittive.

- **Dettagli del prodotto:** sono gli obiettivi dell'interazione successiva. Possono essere storie utenti o epiche descritte in maniera dettagliata, a cui viene associato un ordine di priorità. Nel caso di Galdi, avendo scelto di utilizzare il metodo Scrum, sono gli elementi a cui si lavorerà nello sprint successivo.

Al fine di facilitare la comprensione e la diffusione delle informazioni è utile che il *Product Canvas* venga completato in maniera collaborativa e condivisa da tutti i soggetti interessati dal progetto, in maniera tale da chiarire fin da subito eventuali dubbi e perplessità.

4.1.2 L'uso di Scrum

Scrum è stato scelto come metodologia per gestire la quotidianità del progetto. Si sono seguite le modalità descritte in precedenza nella sezione 2.1.4.2.

A livello aziendale è stato scelto di realizzare sprint della durata di due settimane, in modo da focalizzarsi sulle attività di breve periodo. In Figura 4.2 si può notare la frequenza con cui è stato scelto di svolgere i vari incontri.

Evento	Partecipanti	Frequenza	Durata
Sprint Review	Team di progetto, product owner, scrum master e stakeholder	Ogni 2 settimane	1h / 2h
Sprint Planning	Team di progetto, product owner, scrum master	Ogni 2 settimane	30m / 1h 30m
Retrospettiva	Team di progetto, product owner, scrum master	Ogni 2 settimane	45m / 1h 30m
Raffinamento storie e backlog	Team di progetto, product owner, scrum master	Ogni 4 settimane	1h / 2h

Figura 4.2: Frequenza degli eventi scrum

Fonte: documentazione interna aziendale

Story points Alcuni elementi che meritano un maggior livello di dettaglio, rispetto a quanto già in precedenza spiegato sono collegati alla stima dello sforzo necessario per eseguire i compiti.

Un metodo utilizzato in vari approcci *agile* è quello di assegnare ad ogni storia utente un punteggio in termini di *story points*. Gli *story points* sono un'unità di misura fittizia, appositamente creata per misurare l'avanzamento del team. I punteggi vengono assegnati alle attività sulla base di confronti rispetto ad altri compiti. Secondo Miranda (2001) l'assegnazione di valori tramite confronto consente una maggior precisione delle stime rispetto a quelle effettuate in termini di valore assoluto, che sono soggette a un numero di errori più elevato. Un metodo spesso utilizzato per assegnare questi punteggi è l'uso della serie di Fibonacci. Le storie possono avere un punteggio che prosegue secondo i numeri della serie suddetta (quindi 1, 2, 3, 5, 8 e così via). Il team inizia stimando un compito di dimensioni ridotte, a cui sente di poter assegnare un punteggio con relativa facilità e prosegue quindi, per confronto e seguendo i numeri della progressione, ad assegnare un punteggio ai compiti rimanenti.

La somma degli *story points* da portare a termine entro una determinata iterazione prende il nome di *velocity* del team e rappresenta la capacità del team all'interno di uno sprint. È normale che durante le prime iterazioni ci siano delle fluttuazioni in termini di *velocity*, ma con il proseguire delle iterazioni e la familiarità dell'utilizzo del metodo questi valori dovrebbero stabilizzarsi intorno a una media per uno stesso team. La *velocity* è un elemento caratteristico del singolo team e del singolo progetto e non può venire utilizzata per fare confronti tra team, essendo infatti definita per confronto di compiti, team o progetti differenti potrebbero utilizzare unità elementari diverse.

La *velocity*, tramite l'utilizzo della *burndown chart* illustrata in Figura 4.3, può essere utilizzata per valutare l'andamento di uno sprint in corso ma anche per prevedere il termine del progetto complessivo.

Nel caso della valutazione di uno sprint in corso, vengono confrontati i punti delle storie completate durante lo sprint fino al giorno interessato rispetto a quelli complessivamente pianificati. Il confronto delle curve sarà un indicatore della possibilità di terminare tutti i compiti assegnati entro il periodo prestabilito, oppure se sia necessario accelerare per raggiungere gli obiettivi.

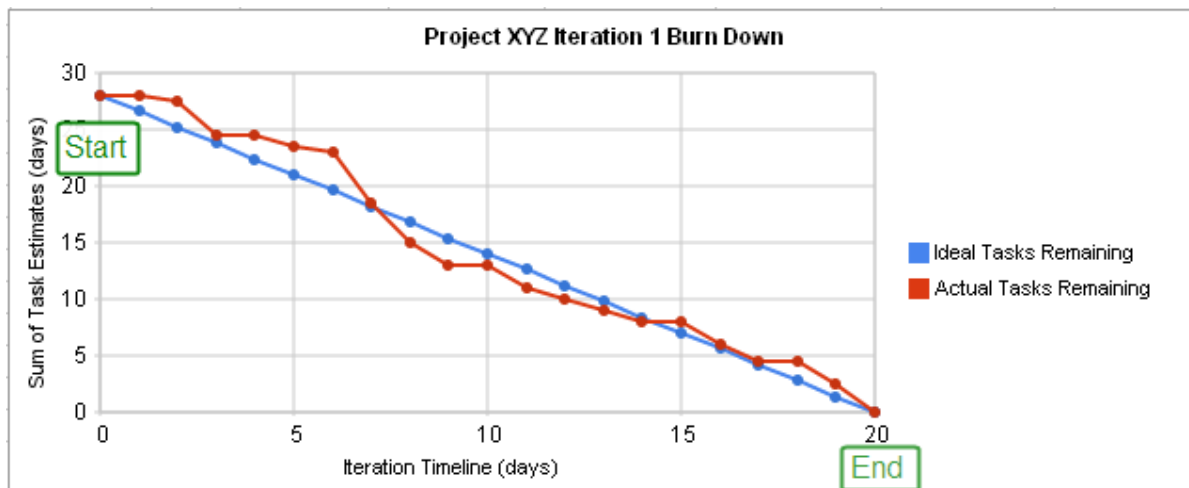


Figura 4.3: Esempio di burndown chart

Fonte: en.wikipedia.org

Rapportando la somma dei punteggi corrispondente alle epiche (o storie) di un progetto ancora da completare con la media della *velocity* (degli ultimi sprint) è possibile calcolare il numero di sprint rimanenti al completamento del progetto.

4.2 Esempio applicativo: il progetto Dry Food

Un mercato che l'azienda ritiene particolarmente innovativo e interessante è quello del cibo secco, o meglio la possibilità di inserire alcune tipologie di prodotti, normalmente confezionati in buste di plastica o in cartoni, nei cartoncini gable top. Questi prodotti possono variare dai cereali, ai semi, alle farine, offrendo ai clienti Galdi la possibilità di differenziarsi dalla concorrenza con un packaging accattivante e funzionale. Da ora in poi ci si riferirà a questa categoria di prodotti con il termine *dry food*.

Già in passato l'azienda aveva svolto dei progetti nel mercato *dry food*. Le macchine, adatte a confezionare questi prodotti, sono state però solitamente fornite senza dosatore, il componente della macchina decisamente diverso rispetto al riempimento di liquidi, che veniva integrato dai produttori di carta o dal cliente stesso. Solo in un progetto l'integrazione del dosatore era stata svolta da Galdi stessa.

L'interesse verso questi mercati è cresciuto negli ultimi anni e l'azienda sta attualmente decidendo di investire in maniera più importante in questo settore, con l'intenzione di fornire ai propri clienti soluzioni complete.

Grazie ad un allineamento con le opportunità di mercato è stato deciso di partire con un primo progetto, in grado di consentire di sondare il mercato.

4.2.1 La storia del progetto e le sue sfide

Lo scopo del progetto era quello di consegnare al cliente una macchina per confezionare diversi tipi di farine, all'interno di un packaging *gable top* conico appositamente studiato, vincitore tra l'altro del premio *ADI Packaging Design Award 2019*.

Il mercato di riferimento del cliente è quello italiano e cercava una soluzione che gli consentisse di formare e riempire quel packaging in tempi molto brevi, visti gli accordi da lui presi con la grande distribuzione.

Nella tabella seguente vengono riassunte le caratteristiche e i principali elementi di sfida del progetto. E' stato scelto di utilizzare le aree definite dal *Project Management Institute* per analizzare il progetto.

Dimensioni chiave	Situazione del progetto	Elementi di sfida
Ambito	Lo scopo di fornitura non era inizialmente ben chiaro, neanche il cliente sapeva bene quali e quanti prodotti voleva confezionare.	I requisiti del prodotto non erano inizialmente chiari e sono emersi con il tempo. Ciò ha reso difficile la pianificazione. La presenza di tre aziende coinvolte ha reso complesso comprendere le specifiche della macchina.
Tempi	I tempi di progetto erano inizialmente troppo brevi per riuscire a realizzare la macchina. Grazie al forte focus sul cliente e all'idea di rilasciare delle versioni incrementali di prodotto il progetto è stato suddiviso in due parti, riuscendo ad incontrare le esigenze del cliente.	I tempi erano molto stretti per entrambi i progetti. Si è cercato di rispettare le tempistiche comunicate al cliente il più possibile.
Costi	Le stime iniziali sono state rispettate.	Il rispetto del budget di progetto ha posto delle limitazioni durante le fasi di sviluppo.
Qualità	Si è cercato di rispettare gli standard di qualità elevati dei prodotti Galdi, anche per la macchina semiautomatica. È stato necessario scendere a dei compromessi per la prima macchina.	È stato necessario svolgere diverse attività per garantire la qualità dei fornitori. Particolarmente delicato è stato il lavoro per il dosatore, che è stato acquistato e sviluppato con un nuovo fornitore, non conosciuto.

Dimensioni chiave	Situazione del progetto	Elementi di sfida
Gestione dell'integrazione	L'integrazione delle varie aree è stata svolta principalmente in maniera informale.	Difficoltà a gestire l'ambito di progetto non definito, le tempistiche brevi e la volontà di mantenere elevati standard di qualità. I numerosi soggetti coinvolti hanno generato problemi e difficoltà nella coordinazione e nell'allineamento delle informazioni.
Gestione delle comunicazioni	La comunicazione è stata svolta principalmente in maniera informale.	La conoscenza tacita e la comunicazione informale hanno reso talvolta difficile recuperare le informazioni.
Gestione dei rischi	La gestione dei rischi è stata valutata in maniera non strutturata mentre il progetto proseguiva.	Non era facile mantenere il focus su tutti i rischi di progetto, non avendo una gestione strutturata. Un forte elemento critico è stato la mancanza della firma del contratto da parte del cliente.
Gestione degli approvvigionamenti	Un elemento critico per il funzionamento della macchina è stato acquistato all'esterno.	Non si conosceva il fornitore. È stato difficile gestire i contatti con il fornitore.
Gestione delle risorse umane	Il team di progetto non era formato da personale esperto e/o dedicato.	L'inesperienza del team poteva rappresentare un forte elemento di sfida.

Dimensioni chiave	Situazione del progetto	Elementi di sfida
Gestione degli <i>stakeholder</i>	I soggetti coinvolti nel progetto erano molti. Erano presenti numerosi intermediari.	Era difficile mantenere un allineamento tra tutti gli <i>stakeholder</i> esterni.

4.2.2 Analisi del progetto

Di seguito verranno trattati alcuni elementi interessanti del progetto in maniera più approfondita, analizzando in particolare l'impatto dato dall'approccio *agile* nella gestione del progetto.

Gestione dell'integrazione

Una delle maggiori sfide, fin da subito evidenti, era rappresentata da un *Time to Market* rapido del progetto, che doveva avere una scadenza di pochi mesi.

Questa è stata una delle prime discriminanti del progetto. Il cliente chiedeva infatti la consegna della macchina entro tre mesi (tempo a cui andavano sottratte le vacanze di Natale che ricadevano all'interno del periodo), decisamente troppo breve per la realizzazione di una macchina completa, considerando le particolarità della richiesta, che non entra pienamente nel business principale di Galdi e richiedeva tempi di sviluppo, ma anche di assemblaggio e collaudo, più lunghi, per la mancanza di conoscenza nelle soluzioni tecniche da adottare.

Grazie alla formazione sui principi *agile* avvenuta in azienda e in particolare al focus sulle esigenze del cliente e alla consegna di soluzioni frequenti è stato pensato di proporre al cliente il noleggio di una soluzione intermedia, un insieme formato da tre macchine, un applicatore tappi, un dosatore e una macchina semiautomatica in grado di compiere due fasi distinte: formazione e saldatura del fondo del cartone vuoto e formazione e saldatura del tetto del cartone riempito, che gli avrebbe consentito di cominciare la produzione e di fornire solo in seguito la macchina automatica completa.

E' stato quindi deciso di suddividere il progetto in due, il primo per realizzare la macchina semiautomatica, chiamato *SAM10*, il secondo per la realizzazione della macchina auto-

matica, detto *RG50 DF*. Ciò, oltre ad aver consentito di rispettare le richieste del cliente ha portato altri vantaggi, in altre aree di conoscenza che saranno spiegati in seguito.

Gestione dell'ambito di progetto

La realizzazione di questo progetto va a inserirsi all'interno di un contesto più grande, che va oltre la consegna di questa singola macchina, ovvero la volontà dell'azienda di investire in macchine in grado di confezionare prodotti *dry food*. Il progetto infatti doveva servire come prova per testare il mercato e le capacità dell'azienda di sviluppare soluzioni adatte.

Sebbene nell'approccio *agile* siano fondamentali il ruolo del cambiamento proattivo e la capacità di adattarsi alle richieste del mercato, i requisiti di progetto sono stati perlopiù statici nel tempo e la filosofia ha paradossalmente aiutato in questo. Gli elementi che hanno contribuito positivamente sono stati la possibilità di sviluppare il prodotto in maniera incrementale e la relazione continuativa con il cliente fin dalle prime fasi.

Gli elementi di incertezza e le richieste di modifica sono stati molti. Sebbene fosse inizialmente presente un'idea di massima del prodotto da realizzare, la configurazione del prodotto non è stata esplicitata in ogni dettaglio e ha subito delle richieste di variazione e ridefinizione nel corso del progetto. Queste incertezze sono state causate sia dal cliente, che non aveva inizialmente ben chiaro cosa desiderasse ma spingeva per ottenere il massimo dalle soluzioni, che dall'azienda e dai numerosi soggetti coinvolti (fornitore di carta in maniera preponderante), che proponevano soluzioni migliori dal loro punto di vista.

In questo senso, la metodologia è stata d'aiuto per mantenere il focus nel valutare con il cliente le richieste di modifica, specificando cosa tali modifiche avrebbero comportato in termini di tempo e costi e lasciando a lui la scelta di realizzarle fin da subito oppure di modificarle in seguito. Il cliente ha deciso di mantenere i tempi di consegna rapidi e realizzare alcune delle modifiche in seguito, come *upgrade* della macchina esistente e valutando concretamente anche l'acquisto di un'altra macchina simile, per altri tipi di prodotti.

La prima grande incertezza era dovuta alla tipologia di prodotti da riempire. Infatti, sebbene le farine fossero state sempre presenti, il cliente voleva valutare la possibilità di riempire prodotti solidi in chicchi, come cereali da colazione. Questi prodotti necessitano di un altro tipo di dosatore, ed è stato per questo garantita la possibilità di rimuovere il

dosatore per sostituirlo eventualmente in seguito.

Altro dettaglio era dovuto alla tipologia di farine che potevano essere dosate in sicurezza compatibilmente alla normativa. Possono infatti creare una nuvola di polvere altamente esplosiva e per questo devono essere trattate in maniera apposita. Durante il corso del progetto è stato stabilito un certo range di tolleranza in relazione alla granulosità della farina.

La richiesta di cambiamento dei requisiti più importante, giunta in una fase molto avanzata della progettazione della macchina *RG50 DF* è stata quella di modifica delle dimensioni dei cartoni. A causa delle modifiche all'altezza degli scaffali della grande distribuzione, i cartoni avrebbero avuto un'altezza troppo elevata, sarebbe stato possibile posizionarli solo nel ripiano più alto e per questo meno appetibile. Per ridurre l'altezza era necessaria una riprogettazione completa del packaging che avrebbe avuto impatti importanti su tutta la macchina. Ciò avrebbe comportato un forte allungamento dei tempi di progetto. La problematica è stata spiegata al cliente, che ha scelto di realizzare comunque la macchina in tempi brevi, in modo da poter adempiere agli impegni presi con i propri clienti, richiedendo però la possibilità di modificarla in seguito.

Un'altra menzione merita la fornitura della linea completa. Inizialmente doveva venire offerta da Galdi la linea realizzata da una società consociata. Non era chiaro in cosa dovesse consistere la fornitura e successivamente è stato messo in discussione anche se venisse fornita da Galdi, direttamente dalla società consociata oppure da terzi.

Infine altre modifiche sono state spinte dal fornitore del packaging preformato. La prima attività infatti è la definizione delle caratteristiche del packaging. Viste anche le sue particolarità in questo caso, comportava alcune scelte progettuali molto importanti e come già detto, la sua modifica comporta modifiche importanti in tutta la macchina. In una fase ormai conclusiva della progettazione era stato proposto la modifica del cartone, per una convenienza economica nella realizzazione. Questo avrebbe comportato serie modifiche alla macchina e per di più, sarebbe stato necessario modificarlo in seguito per andare incontro alle nuove esigenze del cliente di riduzione dell'altezza. La proposta è stata scartata.

Dal punto di vista operativo una sorta di WBS con poco grado di dettaglio viene creata tramite il sistema di individuazione dei Bin. Infatti vengono analizzati tutti i gruppi

funzionali cercando di capire l'entità del lavoro da eseguire.

Nonostante alcune difficoltà, viste le peculiarità necessarie per il *dry food*, che hanno reso necessario adattare le matrici per l'individuazione dei Bin, il processo è stato eseguito per entrambi i progetti e ha portato alla classificazione dei Bin in un S2 per *SAM10* e un M1 per *RG50 DF*.

Per il secondo è stato anche realizzato un *product canvas*, per aiutarsi a inquadrare ancora meglio lo scopo del progetto.

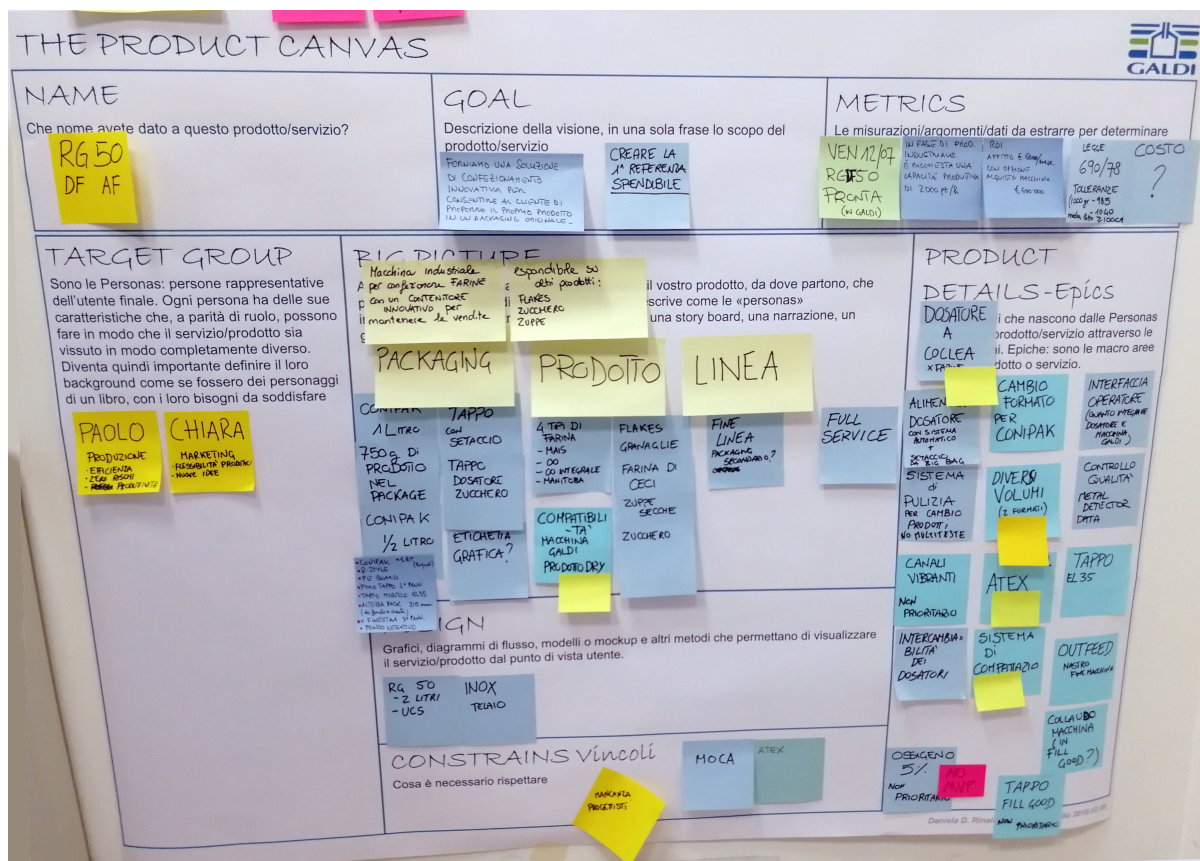


Figura 4.4: Product Canvas RG50 DF

Questi strumenti sono stati utili per definire in una prima battuta i requisiti di progetto e dare una priorità ai gruppi funzionali, cominciando a lavorare fin da subito su quelli considerati più critici perché affrontavano temi non noti, quali l'integrazione del dosatore, lo sviluppo del sistema vibrante, necessario a compattare la farina all'interno del packaging, il cambio formato e la direttiva ATEX.

Seguendo la metodologia *scrum*, la definizione dei requisiti e dei compiti in maniera più dettagliata è avvenuta successivamente con una cadenza ciclica, mano a mano che il lavoro

proseguiva e si ottenevano feedback dal cliente. Ciclicamente è avvenuto anche il controllo e l'aggiornamento degli stessi, rimuovendo attività che si rivelavano non necessarie e definendo nuovamente le priorità.

Un altro elemento che ha causato delle ridefinizioni dell'ambito è stato sottovalutare la complessità di sviluppo e realizzazione del gruppo sistema vibrante. Infatti inizialmente non era stato compreso come il sistema costituisse una vera e propria nuova tecnologia e le tempistiche che richiedesse il suo sviluppo. È stata prevista la realizzazione di test per verificare l'effettiva fattibilità e efficacia delle soluzioni pensate, che hanno richiesto però tempi molto lunghi per venire eseguiti.

Gestione dei tempi di progetto

Come già si è evidenziato nella gestione dell'integrazione, il progetto è stato spezzato e gestito in due. I due progetti sono stati pianificati e gestiti in maniera separata per incontrare le esigenze del cliente.

Per entrambi i progetti è stata realizzata una pianificazione di massima all'inizio del progetto, partendo dalla data di consegna al cliente e proseguendo a ritroso nella pianificazione. La gestione dei progetti è stata *agile* nella parte di progettazione, mentre si è appoggiata ai processi tradizionali nella fasi di approvvigionamento, assemblaggio e collaudo.

Nella parte di progettazione è stato spiegato ai progettisti il lavoro da fare ed è stata lasciata loro la libertà di come gestire il tempo e organizzare i compiti.

Per quanto riguarda il progetto *SAM10*, questo è stato gestito in maniera relativamente strutturata, utilizzando la metodologia *scrum*.

Vista la novità della metodologia questa è stata implementata un po' alla volta, gestendo il progetto con sprint della durata di una settimana, attraverso l'utilizzo di un cartellone *visual* per organizzare i task. Inizialmente non sono stati coinvolti i progettisti, oltre che per la natura particolare del team, anche per non occupare i tempi già molto ristretti in riunioni, e lo scrum master, nonché responsabile del team di progetto, si è occupato di diffondere e condividere le informazioni.



Figura 4.5: Esempio pianificazione settimana

Per il progetto *RG50DF* si sono invece coinvolti tutti i soggetti interessati, svolgendo sprint della durata di due settimane. I compiti da svolgere durante l'intervallo di tempo venivano definiti dai progettisti stessi, con la supervisione del product owner. È stato anche adottato il passaggio a uno strumento di pianificazione sempre visual, ma digitale, per favorire la portabilità e la consultazione delle informazioni.

Le fasi successive si andavano poi a inserire nel normale flusso produttivo, per questo motivo le scadenze dovevano essere più rigide e definite e per questo si sono stabilite delle tempistiche più precise. Infatti sebbene il progetto potesse avere delle scadenze più morbide, perché il cliente era disposto a tollerare qualche settimana di ritardo, ciò non valeva per la normale produzione all'interno del quale si andava ad incastrare, per la quale il mercato richiede consegne precise.

Gestione dei costi di progetto

I costi sono stati gestiti per i due progetti in maniera separata. Per il progetto *SAM10* è stata svolta una stima di massima delle ore di progettazione necessarie e dei costi necessari per le modifiche alla macchina.

Non è stato svolto alcun controllo intermedio e solo alla fine è stata realizzata una verifica dei costi effettivi rispetto a quanto pianificato, che si sono rivelati allineati.

Per il progetto *RG50 DF* le stime sono state fatte per confronto con altri progetti simili, stimando il costo dei dosatori sulla base del costo di quello utilizzato per la *SAM10*. I costi sono stati monitorati ciclicamente, con cadenza mensile, in maniera non strutturata, aggiornando i costi, integrando le aggiunte e rimuovendo ciò che si è rimosso dalla macchina.

Gestione della qualità di progetto

Uno degli obiettivi di progetto era quello di garantire comunque il livello di qualità Galdi.

Gli elementi di incertezza erano molti e per questo il progetto *SAM10* è stato un banco di prova per la macchina finale, in particolare per quanto riguarda il dosatore, e per iniziare a prendere familiarità con i temi specifici del progetto, come le direttive ATEX da rispettare per la possibile formazione di nubi di polveri esplosive. Ha consentito di testare il fornitore del sistema di dosatura, nonché il dosatore stesso, che sono poi stati utilizzati nella macchina finale. Si sono incontrati alcuni elementi di criticità che sono poi stati risolti grazie a modifiche incrementali al dosatore, che hanno portato a raggiungere i requisiti di dosatura richiesti dal cliente, perché obbligo di legge, e garantiti dall'azienda.

Un altro elemento critico, perché nuovo rispetto alle riempitrici di liquido, era costituito dal sistema vibrante, necessario a compattare le polveri una volta inserite nel cartone, presente nella macchina *RG50DF*. Questo elemento è stato approcciato svolgendo alcuni test, fin dalle prime fasi di progettazione, per capire quale sistema utilizzare e i parametri ottimali per il suo funzionamento. L'approccio a questi test, ispirato alla pratica del *test driven design*, ha portato all'esclusione del sistema inizialmente considerato più promettente a favore di un altro più performante nel lungo periodo. Nella realizzazione dei test è stato utilizzato un approccio iterativo, svolgendo dei test incrementali.

L'esecuzione di questi test si è però scontrata con alcune problematiche già note in azienda, quali la mancanza di un processo strutturato per l'esecuzione dei test, in particolare,

la presenza di un'area test poco strutturata, priva di attrezzature, di personale dedicato a svolgere le prove e di una supply chain non abbastanza reattiva nel consegnare il materiale desiderato, che hanno causato un forte ritardo nella loro esecuzione.

A questi si vanno ad associare tutta una serie di processi, normalmente presenti nei progetti aziendali per assicurare un elevato standard di qualità, che sono stati adattati alle particolarità del progetto. Ad esempio è stato contattato un consulente per avere informazioni riguardo alle questioni di sicurezza, non note per il progetto, e sono stati presi degli accorgimenti per il controllo della qualità dei disegni, formando i progettisti e allineandoli agli standard aziendali.

Infine l'azienda prevede una lunga fase di collaudo della macchina finale, che viene testata in funzione, con il prodotto al suo interno.

Se per la *SAM10*, questi sono stati svolti in maniera abbastanza veloce e manuale, per la macchina *RG50DF* si è deciso di eseguirli in maniera più strutturata e vicina all'usuale, testando la macchina nel suo insieme con il prodotto al suo interno. Questo ha richiesto un'organizzazione logistica degli spazi e il noleggio di un sistema di aspirazione polveri.

In realtà durante il progetto ci si è accorti di come, i requisiti richiesti dal cliente fossero inferiori rispetto a quanto offerto normalmente da Galdi. Ad esempio, nella macchina *SAM10*, era presente un lieve difetto nella formazione del fondo del cartone. Nell'ottica di garantire trasparenza, ciò è stato spiegato al cliente che non lo ha ritenuto un problema. Nel mercato liquido, il prodotto non sarebbe così stato accettato.

Gestione delle risorse umane di progetto

Le risorse umane hanno costituito un altro tasto dolente del progetto, non erano infatti presenti le risorse necessarie per realizzarlo. Non erano disponibili i progettisti da dedicare e per questo è stato deciso di formare un team con progettisti dedicati normalmente ad altre attività.

Di fatto, il team era formato da due progettista esperti, ma arrivato in azienda da pochi mesi, che sono stati impiegati completamente al progetto, e un progettista esperto, presente in azienda da diversi anni, che doveva però dedicarsi anche ad altre attività. Questa situazione non era sicuramente ottimale, soprattutto per le attività messe da parte per il progetto e il forte carico di lavoro, cosa che ha creato non pochi malcontenti nell'ambiente

generale.

Le altre figure quali, il progettista software e il progettista elettrico sono stati impiegati a tempo parziale e hanno continuato a dedicarsi ad altri progetti.

Ciò viene riassunto di seguito nella Tabella 4.

Progettista	Background	Tempo dedicato	Criticità
Integratore, progettista meccanico	Progettista esperto, ma nuovo nell'azienda.	Full time	Scarsa conoscenza iniziale delle macchine. Grande pressione per l'enorme carico di lavoro e le molte responsabilità.
Progettista meccanico 1	Progettista esperto, ma esterno all'azienda.	Full time	Rischiava di essere poco coinvolto dal progetto. Non poteva svolgere la validazione dei disegni per politica aziendale, dipendendo da altri.
Progettista meccanico 2	Progettista esperto, con molta esperienza nelle macchine.	Part time	Dedicato ad altre attività. Grande carico di lavoro.
Progettista elettrico	Intermediario interno con esperienza.	Part time	Dedicato ad altri progetti ed altre mansioni, poco tempo disponibile.
Progettista software	Progettista junior	Part time	Poca esperienza nella scrittura di codice. Non molto integrato nel progetto.

Tabella 4: Membri del team di progetto

Il coinvolgimento delle altre risorse (acquisti, controllo qualità, programmazione della produzione), è avvenuto tempestivamente e all'esigenza, non presentando particolari

problemi.

La *SAM10* inoltre, non è stata assemblata come le normali macchine in linea, ma anche a causa del periodo in cui andava assemblata, che capitava durante il pieno picco produttivo, in cui le linee erano completamente sature, è stata costruita e collaudata in un'area a parte, con personale normalmente dedicato ad altro, quali tecnici del service e del software.

La *RG50DF* è stata invece inserita all'interno del normale flusso produttivo, utilizzando le risorse solitamente ad esso dedicate. Questo ha creato alcuni problemi organizzativi, oltre che logistici, specialmente durante il collaudo, per la presenza contemporanea di due prototipi, che lo hanno reso il collo di bottiglia rispetto all'assemblaggio per la normale produzione.

Per quanto riguarda le risorse fisiche, queste sono state gestite nella maniera consueta. Un discorso a parte va dedicato ai componenti necessari per realizzare i test del sistema vibrante. Infatti, sebbene questi siano stati ordinati per tempo, i pezzi sono arrivati in ritardo, compromettendo la realizzazione dei test e obbligando a progettare la funzione per la macchina in parallelo allo svolgimento dei test. Altre accortezze sono state prese per il collaudo della macchina finale.

In generale si è utilizzato un approccio che prevedeva la soluzione dei problemi un po' alla volta, mano a mano che questi si facevano più vicini.

Gestione delle comunicazioni di progetto

La comunicazione è stata svolta principalmente in maniera non strutturata. Il team di progetto era situato in un unico ufficio open space e i progettisti meccanici erano posizionati intorno allo stesso tavolo. Questo ha consentito uno scambio di informazioni efficace, preferendo la comunicazione verbale. Non si sono svolti *daily meeting* strutturati, ma lo *scrum master* ha svolto un grande lavoro per tenere tutti gli *stakeholder* interni informati e aggiornati.

Per quanto riguarda la comunicazione con il cliente, la presenza di un *product owner* molto attivo ha consentito di poter avere spesso le informazioni dal cliente, che veniva contattato di frequente all'esigenza.

L'esecuzione delle *sprint review* infine, ha consentito di mantenere un evento formale, durante il quale era possibile per tutti gli *stakeholder* venire aggiornati nel progetto.

Gestione dei rischi di progetto

Gli elementi di rischio interni al progetto erano molti e sebbene non sia stata svolta un'analisi strutturata redigendo un vero e proprio piano di gestione dei rischi, questi sono stati affrontati e tenuti in considerazione durante tutto il progetto.

Le operazioni di identificazione del rischio, analisi qualitativa, pianificazione e implementazione delle risposte sono state svolte di sprint in sprint, considerando le problematiche più urgenti e imminenti e prevedendo una possibile soluzione e alcuni *piani B*, oltre che identificando quelle future, segnalandole come problematiche, mentre si pensava alle possibili soluzioni.

Come si è potuto intuire dai numerosi elementi di incertezza, anche precedentemente citati, il progetto presentava un elevato grado di incertezza nelle numerose aree.

Uno degli elementi di rischio non ancora citato è stato la mancanza di un ordine da parte del cliente. Nel caso della macchina *SAM10*, il contratto di affitto è stato firmato in prossimità della consegna della macchina, mentre per la *RG50DF* il contratto è stato chiuso in una fase molto avanzata della progettazione, creando anche alcune problematiche per l'ordine dei materiali. Questo rischio è stato accettato dall'azienda, ma nonostante ciò ha rappresentato un forte elemento di stress.

Questo rischio è stato in parte dovuto alla non chiarissima definizione dei requisiti da parte del cliente, che non aveva tutto bene chiaro fin dal principio. A ciò vanno sommati i rischi dovuti allo sviluppo di una macchina con delle tecnologie nuove per l'azienda, in cui quindi era presente il rischio tecnico di non sapere come svilupparle.

Altro elemento è stato la scelta di trascurare altri compiti per portare a termine il progetto. Questo ha causato malcontento nel breve termine, e possibili danni d'immagine nel lungo termine.

È stato inoltre scelto di inserire il progetto all'interno di un flusso produttivo dimensionato per una produzione standard, non per prototipi, mettendo a rischio anche la normale produzione.

I tempi brevi hanno inoltre costretto a disaccoppiare alcune fasi, in particolare la progettazione ha continuato a lavorare anche dopo l'ordine di alcuni materiali, assumendo un certo grado di rischio nella correttezza dei disegni. Inoltre, alcuni materiali sono stati ordinati anticipando gli ordini ai fornitori senza potergli comunicare di che componenti si trattassero, nel tentativo di ridurre i lead time standard di consegna.

Inoltre, il mercato *dryfood* è nuovo per l'azienda, che non conosce quindi le aspettative e gli standard a cui i clienti sono abituati, che sono tra le altre cose sembrati essere più bassi rispetto a quelli del mercato liquido, dove è presente un'elevata competizione e sono richiesti standard molto elevati.

In generale, l'azienda ha riconosciuto il progetto come un progetto particolare ed è stata disponibile ad assumere un grado di rischio relativamente alto, sicuramente maggiore dell'usuale, anche riconoscendo che il progetto potrebbe essere una piattaforma di lancio per molti altri in futuro.

Gestione degli approvvigionamenti di progetto

Per quanto riguarda gli approvvigionamenti di progetto questi sono stati gestiti dagli acquisti con l'aiuto del team di progetto nelle questioni più tecniche.

Un elemento interessante è costituito dal rapporto con il fornitore del dosatore, con il quale si è instaurato quasi una partnership e un rapporto di co-design. I rapporti sono stati gestiti nella parte contrattuale dagli acquisti, mentre nella parte tecnica dal progettista integratore. È stato richiesto un grande sforzo di coordinazione e di investimento per i numerosi feedback necessari.

Oltre a ciò è stato realizzato un accordo particolare per i codici la cui progettazione si è conclusa in ritardo rispetto a quanto programmato. E' stata realizzata una valutazione dei codici che sarebbero stati lanciati in ritardo, valutando la tipologia di lavorazione necessaria e la settimana di assemblaggio, in modo da poter dare un'indicazione ai fornitori di quando saranno ordinati i pezzi e quando è desiderata la consegna, individuando in questo modo quelli maggiormente critici.

Gestione degli *stakeholder* di progetto

L'elevato numero di soggetti esterni coinvolti ha reso difficile la gestione dei rapporti tra tutti.

Dal lato cliente sono nate delle complicazioni a causa della difficoltà di comunicazione e di allineamento tra cliente, azienda e fornitore di carta. Essendo tre i soggetti coinvolti e da allineare, non sempre ciò è stato raggiunto, in particolare con il fornitore di carta. La presenza del *product owner* ha facilitato la comunicazione con il cliente, essendo una persona designata a mantenere i rapporti e a chiedere informazioni. Ciò è stato possibile grazie a una propensione della figura e alla disponibilità dei clienti che erano interessati a avere un continuo scambio di informazioni. Il fornitore di carta invece non si è dimostrato sempre disponibile e talvolta aggiornato sull'avanzamento del progetto, generano anche qualche allarmismo.

Altre difficoltà sono state dovute alla gestione dei fornitori, in particolare quelli che avrebbero dovuto fornire parti integrate alla macchina che dovevano garantire uno scambio di segnali, quali il dosatore e la bilancia. Vista la quantità di informazioni necessarie sia commerciali, che meccaniche, che a livello software è stato delicato gestire i rapporti e è stata richiesta una grande disponibilità da entrambe le parti.

Per quanto riguarda la gestione degli stakeholder interni questa è stata facilitata dai vari incontri previsti da scrum. Il coinvolgimento delle varie funzioni è stato un elemento molto apprezzato da tutti i soggetti, che potevano avere un quadro generale dell'avanzamento del progetto, confrontarsi e ricevere feedback dalle altre funzioni. In generale questa è stata ottenuta non solo grazie agli eventi scrum, ma anche ad appositi incontri organizzati tra il team di progetto e gli altri soggetti (assemblatori, responsabili acquisti, responsabili programmazione produzione).

4.2.3 Analisi elementi *agile*

Come già illustrato, *agile* nasce e si sviluppa nel mondo software e per questo necessità di alcune accortezze per essere adattato al mondo manifatturiero, oltre che le particolari necessità di adattamento proprie dell'industria in cui opera.

Questo progetto rappresenta il primo, insieme ad un altro, in cui viene testata la metodologia *scrum* in azienda. Molti elementi sono stati testati e non sono ancora arrivati in una fase di maturità e l'azienda riconosce delle forti potenzialità di miglioramento.

Un primo elemento ritenuto utile è quello di analizzare come il team e l'azienda si siano interfacciati ai dodici principi dettati dall'*Agile Manifesto*:

Principio	Implementazione	Descrizione dell'applicazione
<p>La nostra massima priorità è soddisfare il cliente rilasciando fin da subito e in maniera continua software di valore.</p>	<p>Implementato</p>	<p>Il cliente è stato posto al centro dell'attenzione. Sono state rilasciate soluzioni intermedie per consentirgli di cominciare al più presto con la propria produzione.</p>
<p>Accogliamo i cambiamenti nei requisiti, anche durante stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.</p>	<p>Implementato</p>	<p>Sono stati accettati alcuni cambiamenti e modifiche nel corso del progetto, facendo però capire al cliente l'impatto delle modifiche che voleva richiedere.</p>
<p>Consegniamo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.</p>	<p>Parzialmente implementato</p>	<p>Vista la natura stessa del prodotto, non è stato possibile realizzare in brevi intervalli di tempo versioni funzionanti dello stesso. Lo sviluppo è stato però realizzato tramite sprint con cadenza di due settimane.</p>
<p>Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.</p>	<p>Parzialmente implementato</p>	<p>La presenza di un <i>product owner</i> ha garantito che venissero rispettate le esigenze del cliente. Il contatto con il cliente è avvenuto a necessità, ma in maniera trasparente.</p>

Principio	Implementazione	Descrizione dell'applicazione
<p>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.</p>	<p>Implementato</p>	<p>Sebbene fossero presenti alcune problematiche nel team (come si è discusso in precedenza) è stata lasciata la massima autonomia organizzativa al team.</p>
<p>Il modo più efficiente e più efficace per comunicare con il team e all'interno del team è la conversazione faccia a faccia.</p>	<p>Implementato</p>	<p>Tutte le relazioni si sono svolte tramite comunicazione diretta, sia con il cliente, sia all'interno del team, che con tutti gli altri <i>stakeholder</i>. Il team lavorava in un ambiente open space, in postazioni prossime gli uni con gli altri, ciò a garantito un ottimo livello di comunicazione. Le sprint review sono state occasione per diffondere le informazioni con tutti gli altri <i>stakeholder</i>.</p>

Principio	Implementazione	Descrizione dell'applicazione
Il software funzionante è il principale metro di misura del progresso.	Non implementato	A causa della natura del prodotto finito, non è stato possibile valutare i progressi sul prodotto funzionante. I test svolti sui gruppi funzionali più critici sono serviti tuttavia a muoversi verso questa direzione.
I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante.	Parzialmente implementato	Sebbene siano state garantite le 40 ore lavorative per tutti, la necessità di dover gestire più progetti in contemporanea è stata per i soggetti interessati un elemento di stress e causando problemi ad altri soggetti dell'azienda.
La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità.	Implementato	L'azienda aveva in precedenza svolto un grosso lavoro per garantire eccellenza nelle fasi di progettazione. Questo è utile a velocizzare le fasi successive.
La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.	Implementato	Si è scelto di mantenere il più semplice possibile il lavoro, in modo da essere veloci e ridurre i costi.
Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano.	Implementato	Il team è stato libero di auto-organizzarsi, riuscendo a realizzare gli obiettivi prefissati.

Principio	Implementazione	Descrizione dell'applicazione
A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.	Parzialmente implementato	Sebbene si sia cominciato a riflettere su ciò in maniera strutturata solo nelle fasi avanzate del progetto, è stato ritenuto utile e di aiuto per migliorare il clima e il lavoro nel suo complesso.

Nonostante le difficoltà iniziali nell'accettare il nuovo approccio, diverso dalle precedenti modalità di lavoro, il team si è rivelato coinvolto nel progetto e favorevole a utilizzare l'approccio proposto in maniera sempre più importante.

4.2.4 Analisi implementazione Scrum

Si seguito viene analizzata l'implementazione della metodologia scrum nel progetto.

- **Scrum team:** il team è stato costituito e ha operato secondo delle modalità coerenti con la metodologia. Il team è mutato nel tempo e ha coinvolto diverse figure, a seconda delle fasi del progetto in cui si trovava. Il core team è stato costituito dai progettisti che hanno potuto auto organizzarsi tra loro e collaborare al fine di raggiungere gli obiettivi di progetto. Il team era collocato nello stesso ufficio open space e i progettisti meccanici erano situati intorno allo stesso tavolo. Ciò ha favorito la formazione del team, la condivisione delle informazioni e l'emergere di soluzioni dei problemi.
- **Scrum master:** il responsabile dell'area di *life cycle management* è stato scelto come *scrum master* del team. Non aveva una formazione *agile* pregressa e si è formato nel tempo mentre il progetto proseguiva. Ha seguito dei corsi di formazione organizzati a livello aziendale ed è stato affiancato da un consulente esperto nel settore. Aveva inoltre esperienze precedenti come *project manager*, dimostrando ottime capacità sia tecniche, che relazionali.
- **Product owner:** il *product owner* era una persona nuova dell'azienda, senza una particolare formazione commerciale. Ha dimostrato però un grande interesse nel progetto e una propensione personale per il ruolo. È stato molto presente e coinvolto,

in maniera forse inusuale rispetto a ciò che ci si aspetta dalla figura, ciò anche per comprendere meglio il funzionamento dell'azienda. Anche grazie alla volontà del cliente, è stato in grado di rappresentarlo, riportando spesso i suoi bisogni e perplessità e avendo una comunicazione continuativa.

- **Product Backlog:** Il *product backlog* è stato costruito dal *product owner* con l'aiuto dello *scrum master*. È stata definita una priorità nelle funzionalità da sviluppare, basata specialmente sulla difficoltà di sviluppo e sulla novità tecnologica di quanto da sviluppare, tenendo presenti anche le indicazioni e i suggerimenti espressi dal cliente. Questa priorità è stata definita in maniera qualitativa definendo una serie di funzioni come maggiormente prioritarie rispetto ad altre. Nella definizione iniziale è risultato di aiuto il *product canvas*, che ha consentito di focalizzare l'attenzione sugli obiettivi del progetto individuando gli elementi più critici e importanti. È stato successivamente spiegato al team di sviluppo, presentando gli elementi da sviluppare e raccogliendo i loro feedback.
- **Sprint backlog:** Non è mai stato realizzato uno sprint backlog in maniera vera e propria. Questo potrebbe essere stato uno degli elementi responsabili delle difficoltà di pianificazione e rispetto dell'esecuzione dei task.
- **Sprint Planning:** Anche il planning dello sprint è avvenuto in maniera poco precisa. È stato effettuato per lo più durante la *sprint review*, recuperando i compiti non portati a termine nella settimana precedente e post-ponendoli a quella successiva. Si è cercato di risolvere il problema, cercando di evidenziare le attività da svolgere, facendo anche qualche timido tentativo di assegnare un punteggio ai vari task per calcolare una certa capacità di esecuzione del team e non sovrastimare le attività realizzabili nello sprint. I punteggi sono stati assegnati in termini di *velocity*, secondo la tecnica della serie di Fibonacci, ovvero assegnando i punteggi ai compiti per confronto. Il sistema è stato provato soltanto per due sprint, incontrando alcune perplessità da parte del team di progetto che non aveva familiarità con l'assegnazione di un punteggio e non di un tempo espresso in ore. Anche la non esplicitazione di alcuni dei compiti svolti dal team per il progetto e l'assenza di uno strumento in grado di tener facilmente traccia di questi punteggi è risultata un deterrente per l'applicazione del sistema e il raggiungimento di una pianificazione soddisfacente.

- **Sprint:** È stato scelto di eseguire sprint della durata di due settimane, allungandoli rispetto a quelli del progetto *SAM10*. Questo per condensare le attività di pianificazione e review e riuscire a effettuare degli incrementi reali nel lavoro. Un elemento positivo è stato le modifiche o gli eventuali cambi nei requisiti nel prodotto sono stati comunicati al team di progetto sono nelle fasi conclusive degli sprint. Questo ha consentito di mantenere alta la concentrazione e di non stressare inutilmente il team.
- **Daily Scrum:** L'attività non è stata effettuata. Le esigenze di allineamento e condivisione di eventuali ostacoli è stata svolta dallo *scrum master* che ha svolto un grande sforzo nel comunicare con i vari membri e aggiornare tutti gli interessati riguardo le informazioni. Anche la locazione fisica (tutti i membri nello stesso ufficio) potrebbe aver contribuito a non far sentire l'esigenza di questo evento.
- **Product Backlog Refinement:** L'attività è stata svolta principalmente dallo *scrum master* in collaborazione con il *product owner* con una cadenza mensile (ogni due sprint), aggiustando i compiti presenti nel backlog di prodotto in modo che riflettesse abbastanza fedelmente lo stato del progetto.
- **Sprint Review:** L'attività è stata eseguita con costanza dopo ogni sprint, mostrando quanto sviluppato fino a quel momento e le soluzioni scelte. Oltre a ciò però, di fatto, una buona parte del tempo destinato alla review veniva dedicato a verificare se i compiti pianificati nello sprint precedente fossero stati effettivamente svolti, trasformandola in una sorta di verifica dello stato di avanzamento del progetto. La partecipazione degli *stakeholder* esterni al core team, è stata saltuaria e sporadica e talvolta sono stati organizzati degli incontri diversi dalla review per spiegare e discutere delle soluzioni con altri membri, riducendo in parte l'utilità dell'evento.
- **Sprint retrospective:** Si è cominciata a svolgere nell'ultimo mese e a partire dagli ultimi incontri anche in maniera strutturata. Sebbene sia in una fase ancora iniziale e non di certo matura, ha portato ad alcuni buoni risultati, sciogliendo alcuni elementi di perplessità presenti all'interno del team e contribuendo complessivamente a migliorare il clima complessivo.

4.3 Valutazione dell'approccio

Di seguito vengono riportate delle considerazioni emerse sia dai feedback delle retrospettive, sia da apposite domande poste allo *scrum master* del progetto.

Gli elementi positivi sono:

- Maggior comunicazione tra reparti.

Questo elemento è stato il grande vincitore nelle retrospettive. La creazione di eventi specifici dedicati a condividere le informazioni ha garantito e facilitato la collaborazione tra tutti i soggetti, anche quelli non dedicati al progetto.

- Condivisione del progetto.

L'elemento è stato apprezzato sia a livello di condivisione delle informazioni tra personale tecnico e *product owner*, quindi vicino al mondo commerciale, sia in quanto ha dato in generale una visibilità al progetto a livello aziendale, che è stato percepito come un progetto di successo, ricevendone un'influenza positiva.

- Possibilità di effettuare test su banchi prova.

Apprezzato soprattutto dai progettisti ha garantito una maggior qualità e una riduzione dei rischi di progetto.

- Gestione dei compiti nello *sprint*

L'organizzazione per sprint di due settimane ha consentito al team di progetto di avere dei compiti definiti da realizzare entro un breve orizzonte temporale. L'assegnazione dei compiti a persone specifiche è stato un altro elemento positivo, che ha garantito che le attività non venissero dimenticate o tralasciate.

- Concetto di *release* incrementali

Il concetto di *release* incrementali, chiamato in azienda *MVP* (*Minimum Viable Product*), è stato di grande aiuto per focalizzare l'attenzione del team sulla realiz-

zazione degli elementi fondamentali per la buona riuscita del progetto, risultando anche uno strumento per gestire le modifiche ai requisiti. Le richieste di modifica venivano analizzate e prese in considerazione in relazione all'*MVP* e alle conseguenze che avrebbero comportato sul progetto. Sono state anche il concetto che ha consentito di realizzare la prima parte del progetto (che prevedeva la realizzazione della macchina semiautomatica).

- Soddisfazione del cliente

Il cliente si è dichiarato soddisfatto della soluzione incrementale proposta, che gli ha consentito di entrare sul mercato in tempi molto brevi, precedendo la concorrenza. Anche la continua collaborazione e lo scambio di idee hanno aumentato la percezione positiva del cliente, che ha così potuto comprendere a pieno il valore del prodotto.

Gli elementi migliorabili sono invece:

- Maggior coinvolgimento delle figure produttive (carpentieri, assemblatori, collaudatori) fin dalle prime fasi di progettazione.

Questo fatto è stato dovuto soprattutto alla mancanza di tempo, sia da parte dei progettisti dedicati al progetto, concentrati nei tempi limitati, sia dalle risorse umane della produzione, saturate dalle normali attività produttive. Non è stato possibile trovare dei momenti comuni in cui condividere le informazioni di progetto. Ciò, unito alla parziale conoscenza del prodotto specifico da parte di alcuni progettisti, ha causato alcune criticità nelle fasi avanzate del progetto.

- Esecuzione pratica dei test di prova.

La mancanza di un'area attrezzata dove realizzare i banchi test e l'assenza di un flusso privilegiato per l'approvvigionamento dei materiali necessari all'esecuzione dei test ha rallentato e ostacolato la loro efficacia.

- Maggior costanza sulla pianificazione di breve periodo.

L'azienda riconosce la possibilità di migliorare ancora nell'applicazione di scrum, definendo e controllando in maniera più accurata gli *sprint backlog* e la definizione del numero di *task* eseguibili all'interno di uno *sprint*.

I benefici riscontrati sono confermati dalla letteratura. Anche nell'analisi realizzata da Gustavsson (2016) la maggior parte dei benefici si possono raggruppare in un aumento della soddisfazione del team e delle performance.

Un elemento particolare è che la metodologia ha di fatto contribuito a mantenere i requisiti più stabili all'interno del progetto. L'elemento può sembrare controintuitivo, ma è stato riscontrato anche nel caso analizzato da Batra (2010).

Di fatto, la suddivisione in sprint e la possibilità di effettuare release incrementali fanno percepire i requisiti come maggiormente stabili durante le singole *release*. Anche il contatto continuo con il cliente e il maggior grado di interazione consentono di spiegare e capire meglio le caratteristiche desiderate nel prodotto, evitando riprogettazioni al termine del progetto e favorendone la soddisfazione.

Altri elementi atipici, particolari del caso Galdi e che si differenziano rispetto all'applicazione standard della metodologia *agile* sono:

- L'impossibilità di rilasciare con frequenza prodotto funzionante.

Questa caratteristica è tipica di prodotti software in cui la progettazione e la scrittura del codice corrispondono alla realizzazione del prodotto finito. In contesti diversi, come in quello manifatturiero, queste fasi sono invece ben distinte e separate temporalmente. Ciò rende impossibile realizzare e testare il prodotto finito in maniera incrementale, che viene realizzato fisicamente solo alla fine del progetto. Nel contesto aziendale considerato, ci si è avvicinati in diversi modi alla realizzazione di questo obiettivo, sia progettando il sistema in maniera incrementale, sia eseguendo dei test sui singoli gruppi funzionali per valutarne in anticipo le funzionalità, sia grazie al concetto di realizzare il prodotto in più versioni, realizzate e consegnate in successione al cliente.

- L'approccio viene utilizzato per cercare di strutturare un processo per lo più de-

strutturato.

Nel mondo software l'approccio *Agile* nasce come strumento per superare modalità di sviluppo troppo strutturate. Nel contesto aziendale il sistema viene usato per strutturare un processo che era inizialmente poco strutturato e a discrezione del singolo team di progetto o del singolo progettista. Questo potrebbe essere anche un elemento positivo, che potrebbe facilitare l'implementazione diminuendo la resistenza all'implementazione, ma rappresenta sicuramente un elemento di differenziazione rispetto alla letteratura.

- La metodologia è stata proposta dai vertici aziendali.

Un altro elemento caratteristico è l'approccio *top down* con cui è stata proposta la metodologia. Questo elemento ha causato delle resistenze iniziali (anche molto forti in alcuni casi) all'applicazione dei nuovi elementi. Uno tra tutti, gli incontri caratteristici di scrum erano considerati troppo lunghi e inutili. Con l'avanzare del progetto però, complice anche la riduzione della durata complessiva degli incontri, questo elemento è stato accettato e ne sono stati riconosciuti i benefici.

5 | Conclusioni

L'obiettivo di questa tesi era valutare la possibilità di applicare un approccio *agile* alla gestione dei progetti all'interno di una realtà differente dal mondo software analizzando i benefici riscontrati. L'azienda di riferimento è una media impresa operante su commessa nell'industria manifatturiera.

In primo luogo è stata svolta un'analisi della letteratura atta a raccogliere i concetti fondamentali circa le metodologie *agile* e la loro applicazione nel mondo software e in contesti differenti, analizzandone punti di forza e di debolezza. Altro obiettivo era valutare gli impatti dell'approccio nella gestione dei progetti.

Questi obiettivi sono stati parzialmente raggiunti: la ricerca ha evidenziato una ricca bibliografia riguardo l'applicazione dell'approccio *agile* nel mondo software, diversi articoli inerenti all'utilizzo dell'approccio e di varianti ibride nello sviluppo di nuovi prodotti, mentre i risultati sono stati carenti per quanto concerne l'impatto di *agile* nella gestione dei progetti.

Per quanto riguarda il caso aziendale gli obiettivi prefissati erano principalmente due: verificare la possibilità di adattare le metodologie proposte, con particolare attenzione alla metodologia *scrum* e analizzarne i benefici ottenuti. L'azienda intendeva risolvere alcune criticità riscontrate in progetti precedenti, quali la difficoltà a gestire le continue modifiche nelle specifiche di progetto e la gestione della pianificazione dei compiti nel breve periodo.

La realizzazione del prodotto finito sviluppandolo tramite interazioni incrementali è stata parzialmente applicata.

Nel software la fase di scrittura del codice corrisponde alla realizzazione del prodotto finito, il software è generalmente suddivisibile in moduli e i moduli sono facilmente integrabili al sistema complessivo.

Per la natura stessa del prodotto realizzato dall'azienda non è stato possibile progettarlo e realizzarlo in maniera incrementale, ma solo la fase di progettazione e la sua pianificazione sono avvenute con questa modalità.

I benefici riscontrati nel progetto dall'applicazione della metodologia sono legati princi-

palmente alla soddisfazione del team, che è stato più partecipe nel progetto, alla maggior condivisione delle informazioni tra tutti i soggetti interessati, alla maggior soddisfazione del cliente e a una maggiore consapevolezza delle tempistiche di progetto.

Gli aspetti negativi sono stati una resistenza alla metodologia da parte del team di progetto, specialmente nelle prime fasi di applicazione, a causa dell'approccio *top down* con cui è stata diffusa la metodologia. Parte di questi elementi di contrasto si è attenuata nel corso del progetto, grazie alla graduale comprensione dei benefici.

L'azienda ha trovato una soluzione alle criticità attraverso l'adozione della metodologia *scrum*. La pianificazione delle attività da svolgere all'interno degli *sprint* e le *sprint review* hanno consentito di ottimizzare la fase di pianificazione nel breve periodo, mantenendo alta l'operatività del team fin dai primi momenti di lavoro. Inoltre grazie a una miglior definizione del *concept* di prodotto, al maggior coinvolgimento del cliente fin dalle prime fasi, alla rappresentazione costante del cliente nella figura del *product owner* e al concetto di rilasciare il prodotto in maniera incrementale è stato possibile avere maggior controllo sulle modifiche ai requisiti di progetto.

Il lavoro svolto fornisce una rappresentazione pratica di una possibile declinazione dell'approccio *agile* in un contesto differente dal mondo software. Può fornire da base per ricerche ulteriori su questi tipi di applicazione, grazie allo sviluppo di ulteriori casi applicativi. Può essere utile anche per altre aziende che intendono affrontare un percorso simile, specialmente nell'ottica di evitare alcune delle criticità iniziali.

Un elemento limitante è stato la dimensione del progetto analizzato, volutamente non di complessità eccessiva. Generalmente infatti un progetto di dimensioni rilevanti ha bisogno di linee guida più definite e potrebbe trarre maggior beneficio dall'utilizzo di approcci ibridi nella gestione di progetti. Anche nel caso dell'azienda in questione, sarebbe opportuno valutare l'impiego di metodologie miste in progetti più corposi, evitando di ricalcare ciecamente l'approccio usato in questo particolare caso.

Un'altra limitazione è l'impossibilità di trarre delle conclusioni generalizzate a partire dal caso applicativo. Oltre ad essere un progetto singolo, l'approccio *scrum* è ancor più influenzato dal team rispetto a un approccio tradizionale: sono i membri stessi del team a dover prendere le decisioni e a organizzarsi il lavoro. Per questo si ritiene che sia difficile

isolare i risultati dovuti alla metodologia da quelli dovuti alle persone che compongono il team.

Elenco delle figure

2.1	Struttura generale del modello di sviluppo <i>waterfall</i>	11
2.2	Ciclo di vita del processo XP	16
2.3	Rappresentazione del <i>framework</i> scrum	18
2.4	Valori <i>Lean Software Development</i>	20
2.5	Metodi usati per la gestione dei progetti nel 2018.	21
2.6	Numero di articoli pubblicati per anno.	22
2.7	Benefici e sfide riportati negli articoli.	22
2.8	Esempio di una <i>kanban board</i>	29
3.1	Schema processo produttivo delle macchine	38
3.2	Tipologie di progetti	41
3.3	Fasi di avvio di un progetto	42
3.4	Fasi della vita di un progetto	46
3.5	Esempio ciclo di test	48
4.1	Template del Product Canvas	52
4.2	Frequenza degli eventi scrum	53
4.3	Esempio di burndown chart	55
4.4	Product Canvas RG50 DF	62
4.5	Esempio pianificazione settimana	64

Riferimenti bibliografici

Agile Alliance, <https://www.agilealliance.org/>

Abrahamsson P., Salo O., Ronkainen J., *Agile software development methods: Review and analysis*, VTT Publications 478:1-112, 2002

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003, May). *New directions on agile methods: a comparative analysis*. In 25th International Conference on Software Engineering, 2003. Proceedings. (pp. 244-254). Ieee.

Awad, M. A. (2005). *A comparison between agile and traditional software development methodologies*. University of Western Australia, 30.

Batra, D., Xia, W., VanderMeer, D. E., & Dutta, K. (2010). *Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry*. CAIS, 27, 21.

Beck K., *Extreme Programming explained: Embrace change*. Reading, Mass., Addison-Wesley, Nov16, 2004

Ciric, D., Lalic, B., Gracanin, D., Palcic, I., & Zivlak, N. (2018, March). *Agile project management in new product development and innovation processes: challenges and benefits beyond software domain*. In 2018 IEEE International Symposium on Innovation and Entrepreneurship (TEMS-ISIE) (pp. 1-9). IEEE.

Coffin R., Lane D., *A practical guide to 7 Agile methodologies*, <http://www.devx.com/architect/Article/32836/0/page/4>, 2006;

Conforto, E. C., Salum, F., Amaral, D. C., Da Silva, S. L., & De Almeida, L. F. M. (2014). *Can agile project management be adopted by industries other than software development?* Project Management Journal, 45(3), 21-34.

Gustavsson, T. (2016). *Benefits of agile project management in a non-software development context-A literature review*. In Fifth International Scientific Conference on Project Management in the Baltic Countries (pp. 114-124).

Keshta, N., & Morgan, Y. (2017, October). *Comparison between traditional plan-based and agile software processes according to team size & project domain (A systematic literature review)*. In 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 567-575). IEEE.

Manifesto for Agile Software Development, 2001, <https://agilemanifesto.org/>

Miranda, E. (2001). *Improving subjective estimates using paired comparisons*. IEEE Software, 18(1), 87-91.

PMI, *PMI's Pulse of the Profession report: Success in Disruptive Times*, 2018.

Project Management Institute, *Agile practice guide*, 2017

Project Management Institute, *A guide to the project management body of knowledge (PMBOK guide)*, Sixth Edition, 2017

Royce, W. W. (1987, March). *Managing the development of large software systems: concepts and techniques*. In Proceedings of the 9th international conference on Software Engineering (pp. 328-338). IEEE Computer Society Press.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). *Challenges of migrating to agile methodologies*. Communications of the ACM, 48(5), 72-78.

Sommer, A. F., Hedegaard, C., Dukovska-Popovska, I., & Steger-Jensen, K. (2015). *Improved product development performance through Agile/Stage-Gate hybrids: The next-generation Stage-Gate process?*. Research-Technology Management, 58(1), 34-45.

Schwaber K. and Beedle M., *Agile Software Development With Scrum*, Upper Saddle River, NJ: Prentice-Hall, 2002

Schwaber K., Sutherland J., *The Scrum Guide*, 2017

Winter, M., Smith, C., Morris, P., & Cicmil, S. (2006). *Directions for future research in project management: The main findings of a UK government-funded research network*. *International journal of project management*, 24(8), 638-649.