

POLITECNICO DI TORINO

Corso di Laurea Magistrale

in

Ingegneria Gestionale



Tesi di Laurea Magistrale

Analisi dei benefici derivanti dall'Implementazione dell'applicativo Cyber Financial con metodologia Agile e introduzione al Knowledge Management

Relatore: Prof. Alberto De Marco

Correlatore: Dott.ssa. Cinzia Divella

Candidato: Alessio Mirizzi

Anno Accademico: 2018-2019

Premessa

La presente trattazione si incentra sullo stage svolto presso Santander Consumer Bank (TO).

Durante il periodo trascorso in questa sede ho avuto l'opportunità di avere un saggio dell'applicazione pratica delle nozioni teoriche acquisite durante il corso di studi, oltre che una condivisione di conoscenze, competenze e abilità derivanti dal lavoro di squadra.

Dedicandomi a questa attività ho avuto modo di acquisire competenze metodologiche, conoscenze concernenti la gestione dei progetti ed alle sue possibili applicazioni a livello manageriale e industriale.

L'obiettivo della tesi è di illustrare, in ambito project management e sviluppo, i caratteri di gestione Agile di un progetto - "WorkFlow Recupero Crediti" – ponendo risalto sulle componenti dello stesso e sulle efficienze rilevate.

Il Metodo Agile, nato in contrapposizione al cosiddetto metodo *Heavyweight*, ha già dimostrato che consente di apportare agilmente modifiche al progetto, di abbattere i costi di produzione, di evitare effort inutili e, soprattutto un eventuale fallimento del progetto.

Tra i metodi Agile più diffusi, particolarmente indicato per i progetti complessi ed innovativi è **Scrum**.

Scrum è un insieme di pratiche che dividono il processo di gestione di un progetto in sprint per coordinare il processo di sviluppo.

È altresì un metodo che si basa sui principi della trasparenza, dell'ispezione e dell'adattamento le cui componenti si dividono in: ruoli, artefatti ed eventi.

L'attuazione di questa metodologia è supportata dall'introduzione di un nuovo sistema di Knowledge management dinamico, univoco ed organizzato, coerente con gli standard attuali aziendale: "Confluence".

L'introduzione di nuovi tool, di un nuovo approccio di gestione agile ha permesso l'analisi dei costi e benefici attraverso KPI di tempo, costi e qualità.

Sommario

Capitolo 1: Introduzione.....	2
1.1 Obiettivo.....	2
1.2 L’Azienda: Santander Consumer Bank.....	4
1.2.1 Santander Consumer Bank in Italia.....	4
1.2.2. Livello di struttura Organizzativa.....	5
Capitolo 2: Teoria Project Management.....	9
2.1 Cosa è il project management?.....	9
2.2 Project Management in the software industry.....	11
2.3 Teoria Waterfall Project Management.....	13
2.4 Teoria Agile Project Management.....	15
2.4.1 Metodologia Scrum.....	17
2.5 Waterfall vs Agile: Pro e Contro.....	23
Capitolo 3: Teoria Knowledge Management.....	26
3.1 Definizione di Knowledge Management.....	26
3.2 Vantaggi che L’ICT ha apportato al KM.....	29
3.3 Principali strumenti ICT a supporto del Knowledge Management.....	31
3.3.1 Opzioni di soluzione basate su cloud.....	33
Capitolo 4: Caso Di Studio.....	37
4.1 Situazione “AS IS”.....	38
4.1.1 Definizioni.....	40
4.1.2 Pianificazione delle attività.....	46
4.1.3 Problemi Riscontrati.....	52
4.2 Situazione “TO BE”.....	54
4.2.1 Definizioni.....	55
4.2.2 Assunzioni.....	59
4.2.3 Sprint Management.....	65
4.2.4. Un Nuovo Modo di gestire la conoscenza: “Confluence”.....	76
Capitolo 5: Analisi dei KPI.....	80
5.1 Tempo.....	81
5.2 Costi.....	85
5.3 Qualità.....	87
5.4 I Benefici del Knowledge management.....	90
Capitolo 6: Conclusioni.....	91
Bibliografia.....	93

Capitolo 1: Introduzione

1.1 Obiettivo

Il lavoro di tesi pone l'obiettivo sulla gestione del progetto "Workflow CBU" in modo Agile rispetto alla metodologia di gestione dei progetti attualmente presente nel contesto aziendale, al fine di trarre benefici dal punto di vista organizzativo e temporale, oltre che ottenendo un notevole risparmio in costi senza impattare sulla qualità.

Tali benefici saranno ulteriormente risaltati dall'introduzione del tool informatico (Confluence) in ottica di knowledge management, grazie al quale si potrà, in maniera efficiente e puntuale, gestire la documentazione attinente al progetto, aggiornandola e modificandola con il coinvolgimento di tutti gli attori del progetto.

Lo sviluppo del progetto nasce dalla volontà della banca di migliorare, in ottica di scalabilità e dinamicità, le strategie di gestione del recupero crediti mediante l'implementazione e l'integrazione dell'applicativo "CyberFinancial".

Per scalabilità si intende: lato tecnico la possibilità di modulare le condizioni delle strategie ed applicarli in maniera immediata, lato utente permette l'analisi della gestione delle pratiche di recupero crediti al fine di efficientare ed automatizzare il processo.

Le strategie ad oggi adottate sino all'introduzione dell'applicativo sono vincolate e subordinate alla possibilità di applicarle e monitorarle. Gli attuali sistemi legacy non consentono dinamicità nell'utilizzo e combinazione ai seguenti campi:

- Prodotto
- Giorni di ritardo
- Scadenza
- Attributo Recupero
- Provincia
- Modalità di pagamento
- Rischio Globale (Scaduto + Capitale A scadere)
- CAP (per le città come ad es. ROMA)
- Classificazione (in casi particolari)

- Tipo Anagrafica
- Attributo Pratica
- Indicazione Rimodulazione
- Emittitore
- Flag Accedo

La situazione AS-IS, ad esempio, non permetteva la possibilità di associare dei pesi alle variabili utilizzate nelle strategie.

Gli strumenti in uso – sempre al momento t_0 - per le strategie di segmentazione del portafoglio sono la matrice di rientro e di smistamento che si occupano, come si può dedurre, di acquisire, elaborare e smistare le pratiche a recupero.

Uno degli obiettivi dell'applicativo è quello di strutturare le strategie non solo in base ai dati attuali della pratica, ma anche in base ad algoritmi applicati e ad esiti di gestione, il cosiddetto *scoring*.

Il sistema di Cyber Financial consente di poter applicare dei pesi alle variabili da noi definite, non solo in base ai dati gestionali e finanziari della pratica al momento dell'analisi ma anche in base al comportamento del cliente (ad es. analisi delle valute di pagamento dei sei mesi precedenti, verifica causale di pagamento per clienti stazionari, stazionarietà della pratica, *etc...*).

1.2 L'Azienda: Santander Consumer Bank

Banco Santander è un gruppo di istituti di credito prevalentemente europei e americani in cui operano filiali sotto il marchio del Banco Santander, per esempio la filiale Santander Consumer Bank in Germania e società esterne acquisite dal gruppo, ad esempio la banca inglese Abbey National PLC poi rinominata Santander UK plc nel 2010.

Santander è costituita da 188.492 persone, 125 milioni di clienti e 3,3 milioni di azionisti. Il servizio core è definito dal servizio della banca al dettaglio che genera il 74% degli utili del gruppo. È la prima banca spagnola, in particolare della zona euro per capitalizzazione e la 15-esima a livello mondiale con 71,16 miliardi di euro.

Il gruppo Santander, ancora, opera in Europa, Sud America, Nord America e Asia, grazie alle sue notevoli e periodiche acquisizioni. Nel 2013 contava oltre 186.000 dipendenti, 14.392 filiali, 3,26 milioni di azionisti e 102 milioni di clienti.

Nel 2013, la società globale Atlantic, insieme a Warburg Pincus LLC, ha acquisito una partecipazione del 50% in Santander Asset Management.

1.2.1 Santander Consumer Bank in Italia

In Italia Santander Consumer Bank è una società leader nel credito al consumo ed è focalizzata su prestiti, cessione del quinto, carte di credito, leasing e conti deposito, oltre ad offrire conti deposito liberi e vincolati per i risparmiatori privati. Grazie a numerose partnership con importanti case automobilistiche e moto, è tra le prime Banche specializzate nel settore dei finanziamenti automotive.

Ha sede in Torino, Corso Massimo d'Azeglio 33/E, ma è presente su tutto il territorio nazionale con circa 200 punti Santander, grazie alle quali, è vicina alle esigenze dei clienti, rendendo il credito facilmente accessibile, secondo i principi aziendali di responsabilità e sostenibilità, e alle esigenze dei dipendenti, grazie alla particolare attenzione alle condizioni di lavoro offerte; nelle politiche di formazione e sviluppo diffuse a tutti i livelli aziendale; nelle strategie di gestione HR che gli sono valse la certificazione di "Top Employers Italia and Top Employers Europe"

1.2.2. Livello di struttura Organizzativa

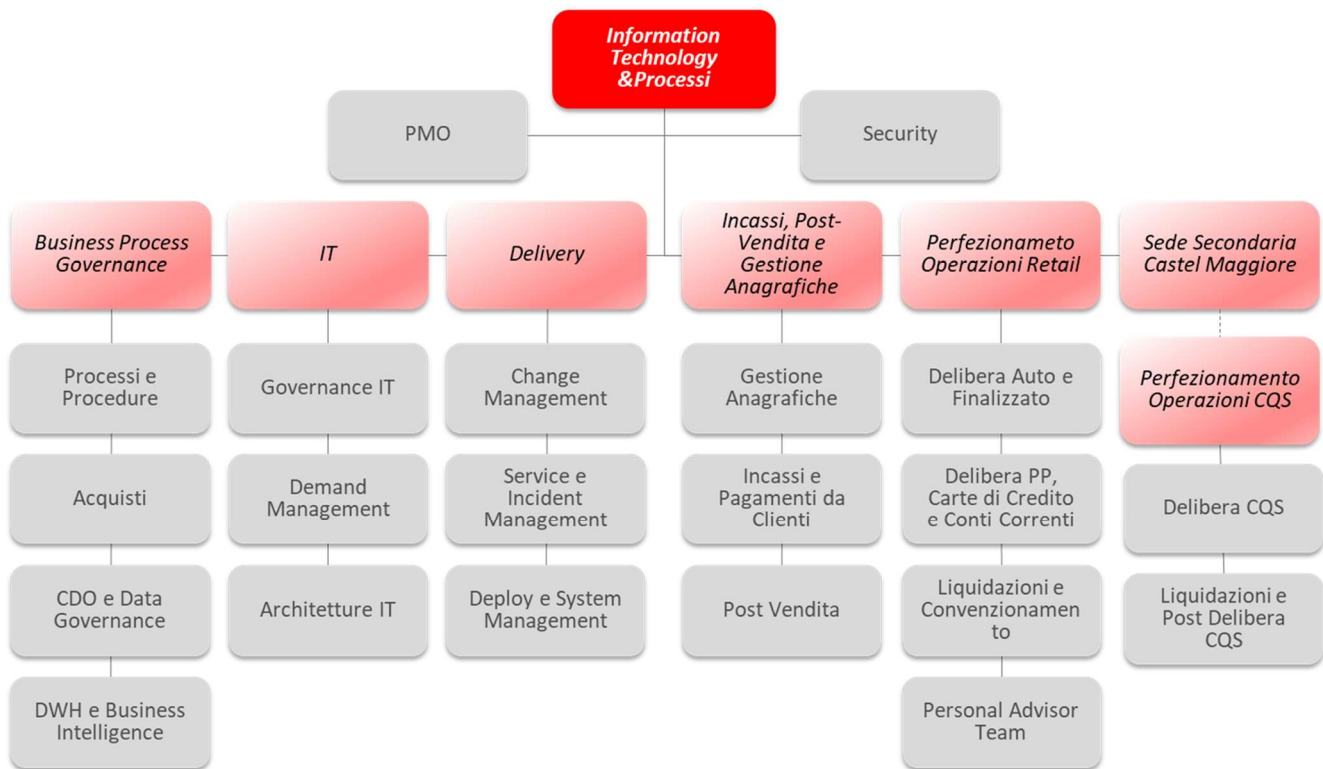
La struttura organizzativa per tutte le società facenti parte del Gruppo Santander Consumer Bank individua, coerentemente con le dimensioni di ciascuna, le seguenti unità organizzative:

- **DIREZIONI**, unità organizzative omogenee per contenuti tecnici ed operativi;
- **SERVIZI**, unità organizzative omogenee per contenuti tecnici ed operativi, che presentano una struttura adeguatamente articolata;
- **AREE**, unità organizzative omogenee per dislocazione territoriale o tipologia di controparte;
- **UFFICI/ UNITÀ LOCALI**, unità operative che raggruppano attività esecutive omogenee.
- **REPARTI**, unità operative in cui possono essere suddivisi gli Uffici.

Per ottimizzare il funzionamento degli organi direzionali sono inoltre formalizzate ove opportuno, delle strutture interfunzionali:

- **COMITATI**: organi permanenti con funzioni consultive e propositive a cui partecipano i Responsabili delle Direzioni, i Responsabili di Servizio oltre a collaboratori interni ed esterni convocati in relazione a particolari problematiche;
- **PROGETTI**: organi operativi transitori costituiti per il raggiungimento di obiettivi specifici non ripetitivi.

Dove viene definito “unità organizzativa” ogni livello della struttura incaricato di svolgere un complesso di attività coordinate per il raggiungimento degli obiettivi aziendali.



1.2.2.1 Direzione Information Technology E Processi

In particolare la mia esperienza di tesi si è svolta all'interno del Servizio Delivery della direzione IT e Processi, la cui mission è: Garantire la migliore gestione della struttura tecnico - organizzativa aziendale. Coordinare le attività a carico dei Servizi che compongono la Direzione, supportare le diverse aree aziendali nelle attività di monitoraggio applicando i modelli corporativi e relazionare sulle attività svolte al Comitato di Direzione, segnalando quelle ritenute strategiche ed il cui impatto a livello aziendale richiede il coinvolgimento di più Direzioni.

Coordinare e supportare inoltre l'attività di gestione dell'IT in coerenza con il system plan, le soluzioni individuate e i budget gestiti dalla Direzione Information Technology e Processi, che ne è funzionalmente responsabile.

Servizio Delivery

Nel dettaglio il Servizio Delivery ha il compito di garantire i livelli di servizio precedentemente stabiliti in merito all'infrastruttura IT e ai livelli di servizio del software.

Per svolgere le proprie funzioni il Servizio Delivery Management si compone di tre uffici: *ChangeManagement, Service e Incident Management, Deploy e System Management*.

La differente gestione delle richieste è descritta in sintesi di seguito e più nel dettaglio nei capitoli successivi dell'elaborazione.

Change Management

Obiettivo principale dell'ufficio è gestire le Change Request e lo sviluppo evolutivo legato ai progetti.

L'ingaggio dell'ufficio avviene mediante lo strumento di ticketing Jira con richieste di tipo CR, utili a tracciare l'attività del gruppo.

Le CR, complete di analisi funzionale e di tutte le informazioni utili allo sviluppo, sono stimate in termini di ore/giornate uomo e restituite all'ufficio demand in modo che possano essere correttamente pianificate.

Il team del gruppo può essere chiamato a partecipare a riunioni preventive all'apertura di CR per dare il proprio contributo per la prima analisi di fattibilità.

Service e Incident Management

Obiettivo principale dell'ufficio è gestire le Service Request ovvero lo sviluppo manutentivo e le configurazioni sistemistiche ed applicative.

L'ingaggio dell'ufficio avviene mediante lo strumento di ticketing Jira con richieste di tipo SR, utili a tracciare l'attività del gruppo.

Le SR, complete di analisi funzionale e di tutte le informazioni utili allo sviluppo, sono stimate in termini di ore/giornate uomo e restituite all'ufficio demand in modo che possano essere correttamente pianificate.

Il team del gruppo può essere chiamato a partecipare a riunioni preventive all'apertura di SR per dare il proprio contributo per la prima analisi di fattibilità.

L'ufficio ha la responsabilità e la facoltà di individuare e proporre sviluppi evolutivi (Change

Request) in modo da ottimizzare e diminuire il livello di configurazione e manutenzione manualesistemistica.

Deploy e System Management

Gli obiettivi principali dell'ufficio sono:

- gestire le release
- gestire l'access management

L'ingaggio dell'ufficio avviene mediante lo strumento di ticketing Jira con richieste di tipo Change Request e Service Request, utili a tracciare l'attività del gruppo.

Il team del gruppo può essere chiamato a partecipare a riunioni preventive all'apertura di CR ed SR per dare il proprio contributo per la prima analisi di fattibilità, con particolare enfasi in ambito networking e sistemistico.

Le implementazioni e le variazioni relative alle aree di networking e Server management saranno concordate e definite in collaborazione con la struttura di Architettura del servizio IT.

Le attività di Access Management saranno svolte secondo le linee guida dell'ufficio Cyber Security.

Capitolo 2: Teoria Project Management

2.1 Cosa è il project management?

Per definire in modo efficace cosa significa "gestione del progetto", è necessario definire prima cosa è un progetto.

Esistono diverse definizioni di progetto:

- Definizione del PMI (project management institute): *“Il progetto è un’iniziativa temporanea intrapresa per creare un prodotto o servizio unico”*.
- Definizione Harvard Business School: *“Un progetto è un’unica serie di attività volta a produrre un risultato definito, con una precisa data di inizio e fine ed una precisa allocazione di risorse”*

Data la natura unica e non ripetitiva di un progetto il rischio e l'incertezza sono due componenti che sono sempre presenti nel lavoro quotidiano.

Per i progetti da realizzare secondo i requisiti stabiliti, nel corso degli anni si è sempre ritenuto necessario avere una figura professionale che assicurasse il rispetto delle scadenze, dei costi e delle condizioni di progetto, il Project Manager.

Lo scopo principale del Project Manager consiste nel cercare di minimizzare questi rischi e le incertezze che inevitabilmente si presenteranno.

Un buon project manager non solo deve conoscere determinate tecniche e metodologie di gestione e prevenzione, bensì deve disporre anche delle cosiddette *soft skills*.

I PM devono possedere capacità comunicative molto forti, devono ispirare fiducia, essere benvoluti, ma allo stesso tempo devono essere forti, carismatici, integri e, soprattutto, credibili.

Il compito principale del PM consiste nel coordinare, in modo efficiente, il lavoro dei vari team di progetto all'interno dell'azienda, garantendo che i requisiti del cliente vengano rispettati.

In particolare, un gruppo di progetto è un insieme di risorse umane che lavora in modo coordinato e cooperativo per svolgere compiti assegnati dal PM, facendo attenzione a rispettare pedissequamente i tempi stabiliti.

A seguire è necessario intraprendere il processo di sviluppo del team di progetto, cercando di motivarlo il più possibile attraverso incentivi e ricompense per coloro i quali otterranno i risultati migliori, assicurandosi che venga rispettata la **regola delle 3C** (Cooperazione,

Coordinamento, Comunicazione). Ad esempio, un certo livello di collaborazione tra i vari componenti del gruppo può essere mantenuto grazie un adeguato coordinamento e sfruttando degli eccellenti strumenti di comunicazione.

Infine, si dovrà procedere con l'effettiva gestione del team di progetto pianificando e allocando le risorse, assegnando a ciascuna risorsa compiti specifici e il più possibile vicini alle proprie capacità e competenze.

In sintesi, un Project Management efficiente dovrà rispettare il cosiddetto “triangolo ferroso”. Esso presenta al vertice lo “Scope”, cioè cosa fare, con quali caratteristiche e funzioni; a destra “Schedule”, cioè entro quale tempo eseguire le varie attività e in quante fasi si divide il progetto; quest’ultimo a sua volta correlato a sinistra con le Resources, numero di risorse necessarie, tipo e costo di ognuna in base al stanziato. (PMI, 2013)



2.2 Project Management in the software industry

Tutto ciò che si è detto nel paragrafo precedente, è legato all'arte della gestione del progetto valido per qualsiasi tipo di progetto: dalla costruzione di un ponte all'implementazione di un servizio. Chiaramente i principi di base e le pietre miliari rimangono i medesimi per ogni tipo di azienda per cui il PM funziona, ma esistono svariate sfaccettature a seconda del tipo di progetto implementato.

L'industria del software è caratterizzata da un alto livello di turbolenza, poiché le aziende IT si trovano a dover adattare i loro servizi o a sviluppare sistemi informativi per organizzazioni complesse, riscontrando difficoltà nella definizione dei requisiti iniziali.

A questo proposito è interessante analizzare il concetto di ciclo di vita del software che si può definire come: “descrizione del software industriale di produzione, dalla sua concezione iniziale fino al suo completo sviluppo, alla sua versione e successiva evoluzione (Pressman, 2008)”.

Quando si realizza un prodotto, è importante seguire una serie di passaggi prevedibili, alla stregua di un percorso guidato che consente di ottenere risultati di alta qualità nel tempo previsto (Su target, On Time, On Budget).

Il termine del ciclo di vita del software indica il modo in cui una metodologia (o un modello) decompone lo sviluppo di prodotti software in attività secondarie coordinate tra loro e, generalmente, include diverse attività:

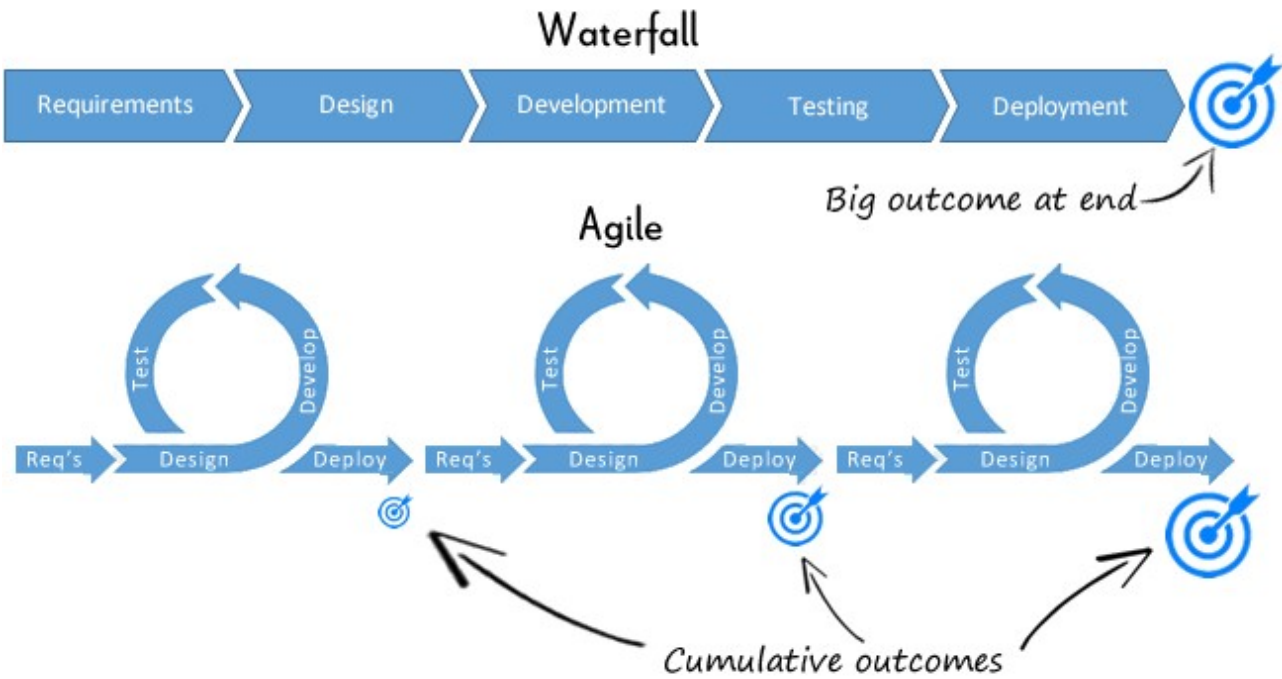
- a. Definizione degli obiettivi, cioè gli obiettivi del progetto.
- b. Analisi dei bisogni e della fattibilità
- c. La concezione generale
- d. Integrazione e test
- e. Rilascio, inteso come la consegna del prodotto al cliente.
- f. Manutenzione

La presenza e la sequenza di ciascuna di queste attività nel ciclo di vita del software dipendono dalla scelta di un approccio o una metodologia di gestione del progetto tra il cliente e il team di sviluppo.

Le categorie di metodologie attualmente presenti sono le seguenti:

1. Metodologie pesanti (come il modello a cascata).
2. Metodologie agili.

Tutti i modelli di sviluppo del software possono includere le attività generiche strutturali descritte in questo paragrafo, ma ognuna di esse applica un'enfasi diversa a ciascuna attività e definisce un flusso di lavoro che coinvolge ogni attività strutturale (e le azioni di ingegneria del software con i loro compiti) in modo diverso ed in momenti diversi.



2.3 Teoria Waterfall Project Management

Il modello a cascata è un modello prescrittivo in base al quale la realizzazione di un prodotto software consiste in una sequenza di fasi ben definita.

Sono indicati come prescrittivi perché prescrivono una serie di elementi del processo: attività strutturali, azioni di ingegneria del software, compiti, risultati, valutazione della qualità, meccanismi di controllo delle modifiche per ciascun progetto, anche stabilendo il modo in cui sono correlati gli uni agli altri.

Il modello a cascata è stato adottato dagli scienziati informatici nei primi anni '70 come primo metodo di lavoro con l'obiettivo di ridurre i costi di sviluppo, rispetto alle pratiche precedenti, e di iniziare a lavorare utilizzando un modello di lavoro ben definito e iterabile.

Il modello a cascata è, probabilmente, il processo di sviluppo software più diffuso al mondo perché richiama la catena di montaggio tipica della produzione industriale.

Suggerisce un approccio sistematico e strettamente sequenziale dello sviluppo del software in cui ogni fase produce un output preciso (deliverable) che viene utilizzato come input per la fase successiva (da cui deriva la metafora della cascata).

Tuttavia, questa metodologia richiede una grande quantità di documentazione, necessaria per informare le persone che lavoreranno al prossimo passo e che non hanno familiarità con i requisiti del progetto.

Le caratteristiche principali di questo modello sono la rigida sequenzialità tra le fasi, che comporta la completa assenza di sovrapposizioni tra di esse, l'assenza di riciclaggio, cioè l'incapacità di tornare a uno degli stadi precedenti per cambiare qualcosa.

Tra le tematiche da tenere in considerazione sono, ad esempio, la *disciplina* (lavorare seguendo regole precise), la *pianificazione* (la necessità di dettare i tempi e rispettarli), la *conoscenza degli obiettivi* prima dell'implementazione del prodotto e, infine, la *divisione in fasi* del lavoro.

L'adozione di questi principi può sembrare estremamente produttiva ma, la loro applicazione, spesso, porta a difficoltà di coordinamento tra le diverse attività (fasi). Supponendo che i requisiti possano essere definiti alla fine della fase di raccolta requisiti, dopo ogni attività deve essere realizzata una corposa documentazione per rendere il lavoro da svolgere nella fase

successiva il più chiara possibile. Definire cosa fare non è sempre facile, questa difficoltà, è una delle principali cause dell'aumento del numero di errori commessi.

Problema di coordinamento evidenziato anche dall'assenza di qualsiasi forma di feedback, sia tra le attività e le persone che svolgono il lavoro, sia con il cliente, con il quale vi è un contatto solo all'inizio del progetto (fase di specifica) e alla fine quando si consegna il prodotto.

Un'ulteriore difficoltà di questo modello è la rigidità della sua applicazione: è necessario completare una fase prima di poter iniziare la successiva.

L'inflessibile divisione del progetto in fasi distinte rende difficile soddisfare le richieste del cliente in merito a modifiche dei requisiti.

Eventuali difficoltà o modifiche ai requisiti comportano ritardi nelle fasi e all'intero progetto, tradotto in maggiori costi di sviluppo e un rinvio del "time to market", uno dei principali problemi in questa metodologia. Il "time to market" è definito come il tempo che può passare dalla commissione di progetto alla consegna di quest'ultimo al cliente.

Nel Waterfall il time to market potrebbe anche durare anni, causando problemi, soprattutto nel settore tecnologico, dove l'obsolescenza rende inadeguato un prodotto anche a breve distanza dalla richiesta di produzione.

Un ulteriore effetto collaterale è l'incapacità di stimare con precisione le risorse e i costi fintanto che non venga eseguita almeno la prima fase di analisi.

2.4 Teoria Agile Project Management

Per questi motivi, l'ingegneria del software ha progressivamente abbandonato l'approccio più tradizionale (Waterfall) a favore dello sviluppo Agile.

Le metodologie agili sono una serie di pratiche di sviluppo software che stanno emergendo a partire dagli anni 2000. L'idea di base non è quella di essere *predittivi*, non si cerca di prevedere come evolverà il sistema, ma *adattivi*, cioè proporre valori e pratiche per adattarsi meglio all'evoluzione costante delle esigenze degli utenti.

Nei progetti software, le esigenze dei clienti cambiano continuamente: sebbene disporre in anticipo di tutti i requisiti degli utenti sia un aspetto desiderabile, spesso non è fattibile. (Bigatti, 2002).

"Tutto cambia nel software. I requisiti cambiano. Cambiamento di design. Il business cambia. La tecnologia cambia. La squadra cambia. I membri del team cambiano. Il problema non è il cambiamento, di per sé, perché il cambiamento sta per accadere; il problema, piuttosto, è l'incapacità di far fronte al cambiamento quando arriva ". (Kent Beck)

Di fronte a situazioni turbolente, in cui il mercato o la tecnologia cambiano rapidamente e sono difficili da prevedere, è necessario adottare un approccio completamente diverso per la gestione dei progetti.

In quest'ottica, quindi, molto importanti sono i concetti di iterazione e feedback continuo che ben si adattano alle dinamiche moderne. Ogni iterazione ha una durata limitata, generalmente da una a quattro settimane. In gergo si dice che ogni iterazione è "timeboxed" e può essere immaginata come un piccolo progetto da realizzare, in cui sono presenti le normali fasi del processo di sviluppo del software, con la differenza, rispetto la vecchia metodologia, che c'è una risposta immediata da parte del cliente, soprattutto data la stretta collaborazione tra il team di sviluppo e gli esperti del business (uno dei principi fondanti definiti da Beck nel Manifesto Agile).

La sintesi delle principali tecniche utilizzate da questo approccio sono:

- Pianificazione: la pianificazione non viene calcolata sulla base di una ripartizione dettagliata di tutte le attività pianificate che vanno dall'inizio alla fine dell'intero progetto ma, al contrario, una pianificazione dettagliata viene sviluppata per un futuro

prossimo a breve termine (4 settimane). Inoltre, la pianificazione viene aggiornata a intervalli regolari (1/2 settimane) che prendono il nome di sprint.

- Pianificazione flessibile: le attività sono classificate in base alla loro incertezza e creatività.
- Gestione del tempo: con la gestione dei tradizionali progetti a cascata, il project manager doveva garantire che i risultati fossero raggiunti gestendo un certo tempo target già impostato ex-ante. Con il metodo agile viene eseguito l'approccio opposto, il progetto viene suddiviso in intervalli regolari (sprint) e si identificano i risultati attesi alla fine di ciascuno. Nessun ritardo è consentito durante l'esecuzione e ogni team di progetto deve segnalare i risultati raggiunti (revisione sprint).
- Specifiche: le specifiche evolvono nel tempo con un numero elevato di iterazioni al rilascio di ciascuna nuova soluzione, ciascuna delle quali consente, a ciascun membro del team, di avere una comprensione più profonda del problema e delle specifiche (Cantamessa, Montagna, 2016).

Per avere una visione ancora più chiara ed esplicita, è interessante analizzare le differenze sostanziali tra questo metodo innovativo e quello più tradizionale, a cascata, prendendo in considerazione lo strumento più significativo di gestione del progetto, il cosiddetto triangolo ferroso. È bene notare come l'importanza attribuita a ciascuno dei cambiamenti dei vincoli chiave, si avvalga di questi due diversi approcci.

Nella cascata tradizionale le caratteristiche, o requisiti, sono mantenute fisse, chiare e ben definite, procedendo con lo sviluppo del progetto solo dopo aver effettuato la stima del costo e del tempo. Dall'altra parte, nella metodologia agile, si può osservare un cambiamento “brusco” e meccanicistico inerente lo scopo e le funzionalità, mentre il costo e le fasce orarie da seguire sono mantenute stabili.

Se nel modello a cascata è la pianificazione che guida l'intero progetto, nel modello agile lo scopo principale è quello di mantenere una buona qualità attraverso l'iterazione fissa in time box, modificando anche i requisiti. (Gerard, 2015).

2.4.1 Metodologia Scrum

Il funzionamento di Scrum è riportato in uno schema del flusso delle attività e dei processi della metodologia Scrum, come definito nello “Scrum Body Of Knowledge”.

Il flusso della metodologia Scrum è così costituito:

Fasi

a) Inizio

1. Creare la vision del progetto: In questo processo viene analizzato il business case del progetto per creare un documento ufficiale che dichiara quale sia la vision (Project Vision Statement) che verrà seguita durante tutto il ciclo di vita del progetto. In questa fase viene identificato il Product Owner;
2. Identificare lo Scrum: Master e gli stakeholder. In questa fase viene identificato lo Scrum Master sulla base di alcuni criteri di selezione;
3. Creare lo Scrum Team: In questo processo il Product Owner e lo Scrum Master identificano i membri del team;
4. Scrivere le Epic: Servendosi del Project Vision Statement, vengono scritte delle User Story di alto livello chiamate Epic. Queste sono troppo ampie per essere completate dal team in un solo Sprint e, per tale ragione, vengono scomposte in User Story più piccole;
5. Creare il Prioritized Product Backlog: In questo processo le Epic vengono raffinate e ordinate secondo un ordine prioritario per creare il Prioritized Product Backlog del progetto. Inoltre, vengono stabiliti i criteri del cosiddetto “Fatto”;
6. Stilare il Release Planning: In questo processo vengono revisionate le User Story contenute nel Prioritized Product Backlog per realizzare il Release Plan (piano di pianificazione dei rilasci), una schedulazione condivisibile con gli stakeholder. In questa fase viene anche determinata la lunghezza degli Sprint.

b) Pianificazione e stime

1. Scrivere le User Story: in questo processo il Product Owner, con l'aiuto del team, scrive le User Story e ne definisce i criteri di accettazione. Queste devono essere progettate garantendo che i requisiti dei clienti vengano rispettati e compresi da tutti gli stakeholder. Le User Story vengono poi inserite nel Prioritized Product Backlog;

2. Approvare, stimare e affidare le User Story: in questo processo le User Story per uno Sprint vengono approvate dal Product Owner e successivamente lo Scrum Master e il team effettuano una stima dello sforzo necessario per sviluppare le funzionalità descritte nelle User Story;

3. Creare la lista delle attività: durante il Task Planning Meeting le User Story approvate e stimate sono suddivise in attività;

4. Stimare le attività: durante il Task Estimation Meetings il team effettua una stima dello sforzo richiesto per realizzare le attività presenti nella lista (Effort Estimated Task List);

5. Creare lo Sprint Backlog: in questa fase il team conduce lo Sprint Planning Meeting, in cui viene creato lo Sprint Backlog, ovvero la lista delle attività da realizzare durante lo Sprint.

c) Implementazione

1. Creare i deliverable: in questa fase il team lavora alle attività presenti nello Sprint Backlog per creare i deliverable dello Sprint. Per monitorare il lavoro svolto e quello ancora in sospeso, si utilizza una Scrum board. I problemi risolti, o da affrontare, sono registrati su un diario chiamato Impediment Log;

2. Condurre i Daily Standup Meeting: ogni giorno sono condotti dei meeting di durata stabilita (es. 15 minuti) che sono utili per i membri del team perché possono sfruttare l'occasione per aggiornarsi sui diversi progressi e impedimenti;

3. Portare avanti il Prioritized Product Backlog: in questo processo il Prioritized Product Backlog viene continuamente aggiornato ed incrementato. Inoltre, si può tenere un Prioritized Product Backlog Review Meeting, in cui eventuali modifiche o aggiornamenti al Backlog sono discussi e, eventualmente, incorporati.

d) Revisione e retrospettiva

1. Convocare lo Scrum of Scrums: solo per i progetti di grandi dimensioni in cui sono coinvolti più Scrum Team, i rappresentanti del team convocano gli Scrum of Scrums Meeting, a intervalli prestabiliti o quando richiesto, per collaborare e tenere traccia dei rispettivi progressi, impedimenti e delle dipendenze tra i team;

2. Dimostrare e validare lo Sprint: in questo processo, durante lo Sprint Review Meeting il team dà dimostrazione dei deliverable prodotti durante lo Sprint al Product Owner e agli

stakeholder più rilevanti, al fine di garantire l'approvazione e l'accettazione da parte del Product Owner;

3. Condurre la retrospettiva dello Sprint: in questo processo il team e lo Scrum Master si incontrano per discutere delle lezioni imparate durante lo Sprint.

e) Rilascio

1. Consegnare i deliverable: in questo processo i deliverable accettati sono consegnati agli stakeholder. Un documento formale di consegna dei deliverable funzionanti attesta che lo Sprint si è concluso con successo;

2. Condurre una retrospettiva del progetto: in questo processo, che conclude il progetto, gli stakeholder e lo Scrum Team si incontrano per discutere di come è andato il progetto e delle lezioni apprese.

I ruoli

a) Product Owner.

Secondo lo Scrum Body of Knowledge e La Guida a ScrumTM, il Product Owner:

- Rappresenta gli interessi della comunità degli stakeholder di fronte allo Scrum Team e lavora con il team, ricoprendo così un ruolo di interfaccia tra il business e il team;
- Ha la responsabilità di gestire il Product Backlog, di cui esprime e ordina secondo priorità gli elementi, decidendo quindi le funzionalità del software ed eventuali modifiche;
- Assicura che il team comprenda gli elementi del Product Backlog al livello necessario;
- Assicura la consegna di valore da parte dello Scrum Team, decidendo sulle date e sui contenuti dei rilasci;

b) Scrum Master

Secondo lo Scrum Body of Knowledge e La Guida a ScrumTM, lo Scrum Master:

- È il “servant leader”, cioè un leader al servizio dello Scrum Team. Impersona il ruolo di leadership collaborativa il cui compito è quello di monitorare e facilitare le relazioni del team, fungendone da maestro e motivatore.

- È responsabile della gestione del processo e, quindi, deve fare rispettare i principi, le pratiche e le regole di Scrum. Inoltre, assiste il team in ambienti organizzativi in cui lo Scrum non è ancora pienamente adottato e compreso;
- Mantiene la documentazione richiesta;
- Aiuta l'organizzazione guidandola nell'adozione di Scrum, pianifica la sua implementazione e aiuta dipendenti e stakeholder a comprendere, ed attuare, tale metodologia di lavoro.

c) Scrum team

Secondo lo Scrum Body of Knowledge e La Guida a Scrum™, lo Scrum Team:

- Sviluppa il prodotto o servizio e quindi viene a volte chiamato Team di Sviluppo;
- È costituito da cinque – dieci persone tra cui sviluppatori, business analyst, tester. La dimensione ottimale del Team di Sviluppo è abbastanza piccola da rimanere agile e abbastanza grande da completare un lavoro significativo all'interno dello Sprint
- È auto-organizzato e nessuno, neanche lo Scrum Master, dice al team come trasformare gli elementi del Product Backlog in incrementi di funzionalità potenzialmente rilasciabili;
- È cross-funzionale e, nonostante i singoli membri possono avere competenze specialistiche e aree di specializzazione diverse, la responsabilità deve essere imputata al team nel suo complesso. La sinergia risultante ottimizza l'efficienza e l'efficacia complessiva del team;

Gli eventi di Scrum

Gli eventi prescritti sono utilizzati in Scrum per creare regolarità e ridurre al minimo la necessità di incontri non definiti in Scrum. Tutti gli eventi sono limitati temporalmente, così da avere una durata massima fissa, ma se il loro scopo è stato raggiunto possono terminare in anticipo. Di seguito sono riportati i principali eventi:

- User Group Meeting: Si tratta di una o più riunioni che si possono tenere durante la fase di scrittura delle Epic e coinvolgono gli stakeholder del progetto, lo Scrum Team e gli utenti e i clienti del prodotto. Scopo di queste riunioni è quello carpire informazioni circa le aspettative e i bisogni degli utenti. Ciò sarà di aiuto per formulare le Epic e i relativi criteri di accettazione;
- Task planning meeting: Si tratta di una o più riunioni che coinvolgono lo Scrum Team e il Product Owner utili per identificare quali User Story inserire nello Sprint. Dietro

suggerimento del Product Owner, ed in base alla capacità del team, vengono concordate le User Story da sviluppare che saranno poi raggruppate in attività da parte del team;

- Task Estimation Meeting o Sprint Planning Meeting: durante i Task Estimation Meeting, o Sprint Planning Meeting, il team stabilisce l'obiettivo dello Sprint (chiamato Sprint Goal), valuta cosa può essere consegnato nell'incremento risultante ed effettua una stima dell'effort necessario per sviluppare le attività identificate. A queste riunioni partecipano lo Scrum Team, il Product Owner e lo Scrum Master e possono essere combinate al Task Planning Meeting;

- Sprint: Lo sprint è l'iterazione in cui il team lavora ai deliverable che andranno a costituire un incremento del prodotto potenzialmente rilasciabile. Si tratta di un periodo di durata limitata che può durare da una a sei settimane. È consigliabile non andare oltre questo limite temporale in quanto, su periodi troppo lunghi, potrebbe cambiare la definizione di ciò che si sta progettando o potrebbero crescere complessità e rischi.

- Daily Standup Meeting: si tratta di riunioni quotidiane in cui lo Scrum Master e lo Scrum Team discutono in maniera concisa dello stato di avanzamento delle proprie attività. Per convenzione ci si pone tre domande:

o Cosa ho fatto ieri che ha aiutato il team a raggiungere lo Sprint Goal?

o Cosa farò oggi per aiutare il team a raggiungere lo Sprint Goal?

o Quali difficoltà sono emerse?

Al termine del Daily Standup Meeting i membri discutono dei punti emersi ed, eventualmente, delle rilavorazioni necessarie, andando quindi ad implementare il principio dell'accoglienza del cambiamento e dell'adattamento tipico delle metodologie iterative;

- Sprint Review Meeting: è una riunione che si tiene a conclusione di ogni Sprint in cui il team presenta al Product Owner e agli stakeholder l'incremento realizzato, al fine di fare emergere eventuali feedback e domande in merito per le implementazioni future.

- Scrum Retrospective Meeting: è una riunione che si tiene dopo lo Sprint Review Meeting e prima del successivo Sprint Planning Meeting a cui partecipano lo Scrum Team, lo Scrum Master, il Product Owner e gli stakeholder del progetto. In questa occasione i partecipanti possono liberamente parlare delle esperienze e dei problemi che hanno dovuto affrontare durante lo Sprint in merito alle persone, alle relazioni, ai processi e agli strumenti utilizzati, al fine di identificare quali aspetti migliorare per gli Sprint successivi.

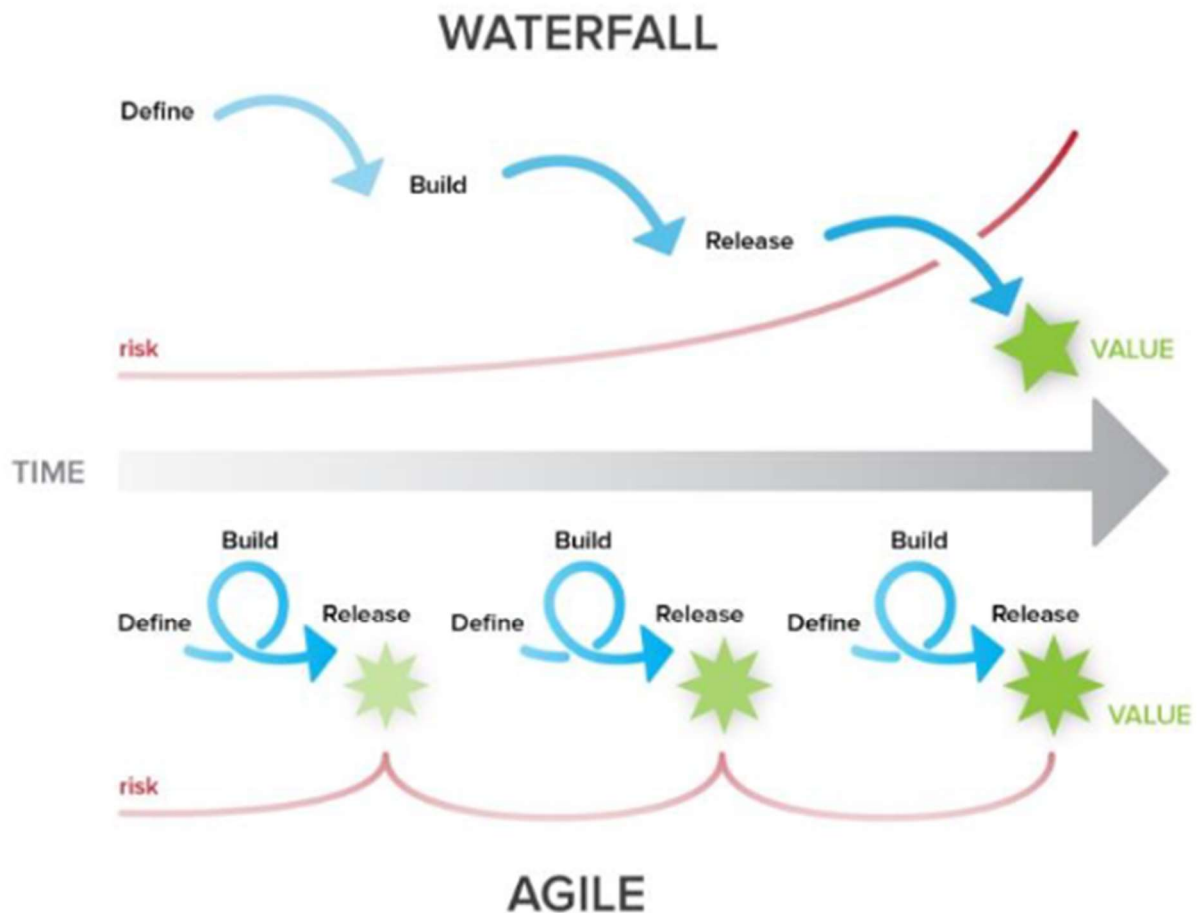
Gli artefatti di Scrum

Scrum e in generale le metodologie Agile, sono caratterizzate da una produzione molto scarna di documentazione. Il termine “Lightweight” sta infatti ad indicare come queste metodologie siano “leggere” anche in termini di documentazione prodotta. Per convenzione, i documenti di Scrum prendono il nome di “artefatti”. Gli artefatti sono progettati per garantire i principi di trasparenza, ispezione e adattamento, in modo che tutti ne possano avere una chiara comprensione e che tutti possano contribuire al loro miglioramento.

- **Product Vision Statement:** il Product Vision Statement è un documento che comunica la vision e gli obiettivi finali del progetto e del prodotto in un’ottica maggiormente strategica e di alto livello per l’impresa.
- **Product Roadmap:** la Product Roadmap è una visione di alto livello delle funzionalità richieste per raggiungere la vision del progetto. Fornisce al team una visione ampia del contesto.
- **Product Backlog:** il Product Backlog è una lista di tutti i requisiti, le caratteristiche e le funzioni che il prodotto dovrà avere scritti sotto forma di User Story e ordinati dal Product Owner secondo regole di priorità da lui definite. I suoi elementi hanno i seguenti attributi: descrizione, ordine, stima e valore e spesso includono le descrizioni dei test che provano che gli elementi sono "Fatti".
- **Sprint Backlog:** lo Sprint Backlog è l’insieme degli elementi del Product Backlog selezionati per lo Sprint ovvero è la lista del lavoro che lo Scrum Team deve fare durante lo Sprint per costituire il prossimo incremento. Durante il corso dello sprint non è possibile apportare modifiche ai requisiti presenti nello Sprint Backlog, tuttavia, il team, se lo ritiene necessario, può aggiungere del nuovo lavoro o rimuovere gli elementi ritenuti inutili. È di esclusiva appartenenza dello Scrum Team. L’incremento è un insieme degli elementi del Product Backlog completati durante lo Sprint che rispetta la definizione di “Fatto” data dal team. L’incremento risultante da uno Sprint deve, quindi, essere utilizzabile indipendentemente dal fatto che verrà poi rilasciato in seguito alla decisione del Product Owner.
- **Burndown chart:** il Burndown Chart è un grafico che mostra chiaramente quanto lavoro dello Sprint Backlog ancora rimane da sviluppare.

2.5 Waterfall vs Agile: Pro e Contro

Nel seguente paragrafo risulterà utile definire nel dettaglio le caratteristiche delle due metodologie evidenziandone vantaggi e svantaggi di uno e dell'altro.



- **Waterfall**

L'approccio a cascata durante il suo lungo periodo di utilizzo in diversi campi, ha permesso di evidenziare numerose caratteristiche, sia positive che negative:

Pro:

- a) I requisiti, comprese le funzionalità di base come tempi e costi, sono ben definiti e concordati sin dall'inizio del progetto;
- b) I limiti di tempo rigorosi e l'ampia documentazione prodotta consentono un monitoraggio e un controllo più facile in termini di sforzo, oltre che un processo di manutenzione più semplice, una volta che il sistema IT risultante inizierà a essere utilizzato;

- c) L'elevato dettaglio delle diverse fasi descritte dalla documentazione consente di impiegare lavoratori meno qualificati per i lavori.

Contro:

- a) L'approccio waterfall può richiedere molto tempo per lo sviluppo completo, specialmente per la lunghezza che i processi di analisi e pianificazione possono raggiungere;
- b) I requisiti, una volta che sono stati decisi e impostati, possono essere modificati solo con processi estremamente difficili, perché potrebbero implicare il ritorno a fasi precedenti e già completate;
- c) La scarsa comunicazione, data dall'esistenza di fasi distinte, che di solito coinvolge persone diverse, può portare a problemi rilevabili solo nella fase di test che potrebbero essere risolti solo tornando a fasi molto distanti, a discapito di un enorme aumento dei costi. Questo si traduce in una notevole pendenza (verso l'alto) durante tutto il progetto, con un picco profondo durante gli ultimi passaggi.
- d) Il cliente di solito vedrà i risultati finali solo alla fine del progetto e questo potrebbe far nascere vari problemi tra cui: la non più attualità del prodotto, bassi livelli di soddisfazione o richieste di modifiche che potrebbero portare a problematiche inaspettate, specialmente in termini di tempo e denaro da impiegare.

- **Agile**

La caratteristica comune tra queste diverse metodologie è lo sviluppo incrementale del sistema target, ottenuto dal rilascio di set consecutivi di software funzionante e potenzialmente rilasciabili, spesso organizzati in timebox relativamente brevi e molto focalizzati sull'ambito. In questo modo, la catena del valore "raccolta delle specifiche → sviluppo software → feedback degli utenti" viene abbreviata il più possibile.

Poiché il puro framework Agile non è applicabile a tutti i tipi di progetti da sviluppare (e anche particolarmente difficili da implementare per ambienti altamente regolamentati), sono nate metodologie ibride per far fronte alle diverse esigenze delle varie aziende. È un dato di fatto, il Manifesto non descrive un modello statico da seguire rigorosamente, ma consta di una serie di suggerimenti e punti di vista raccolti da persone esperte con l'obiettivo di migliorarne l'ambiente ed i processi.

Le caratteristiche principali delle due facce della medaglia del paradigma Agile sono:

Pro:

- a) Ciò che è stato sviluppato è facilmente controllabile dalle parti interessate ed è possibile fornire un numero di feedback molto superiore rispetto un approccio tradizionale.
- b) L'elevata comunicazione richiesta significa, nella maggior parte dei casi, un importante impulso al morale del team di sviluppo. Ciò si riflette anche in una migliore qualità del prodotto finito, visto lo sforzo maggiore posto in esso.
- c) Le metodologie agili sono altamente raccomandate quando le caratteristiche del prodotto finale non sono molto chiare, anche per il cliente, nelle prime fasi di contratto, situazione non così insolita. Se gli elementi diventano più chiari una volta avviato il progetto, verranno inseriti con maggiore o minore facilità in una delle diverse iterazioni.

Contro:

- a) Soprattutto per il vertice strategico e di alto livello, vista l'alta auto-organizzazione solitamente adottata dai team Agile, può esserci una percezione di non controllo e che può portare a contrasti, specialmente con i clienti.
- b) Nonostante la sua adattabilità, i metodi Agile portano incertezza su quale sarà il prodotto finale in termini di funzionalità, ma anche di costi e tempi. Il cliente che investe i suoi soldi in questi progetti, di solito, vuole sapere cosa sta per ottenere e questo può portare alla nascita di contenziosi.
- c) Nonostante la sua apparente facilità di comprensione, i metodi Agile possono essere difficili da applicare efficacemente. Ad esempio, per applicare correttamente il framework è altamente raccomandata la presenza fisica dei soggetti coinvolti. Questo può essere difficile per il team quando il progetto è sviluppato in più di un paese o nel caso in cui i rappresentanti per le esigenze del cliente, figure importanti nello sviluppo Agile, non siano disponibili, per motivazioni quali: mancanza di tempo o impegno. Inoltre, può essere difficile, specialmente durante i primi tempi di adozione delle metodologie, trovare persone che le applichino correttamente, impegnandosi a rispettare la nuova metodologia lavorativa.

Capitolo 3: Teoria Knowledge Management

3.1 Definizione di Knowledge Management

Il knowledge management è la risposta manageriale alla crescente importanza che riveste la conoscenza come fattore competitivo che, essendo una risorsa critica per l'impresa, ha bisogno di essere gestita in maniera consapevole e sistematica.

Non è facile dare una definizione univoca del knowledge management, perciò è opportuno vederne diverse per capire bene il concetto:

- *“il Knowledge Management consiste in programmi ed interventi volti a favorire i processi di creazione, memorizzazione, diffusione della conoscenza combinando interventi sulle persone, sulla struttura organizzativa, sui sistemi informativi, sulla cultura e sul clima organizzativo”.* (Casalino 2012)
- *“il KM è la disciplina manageriale che studia la conoscenza aziendale e che si occupa di individuare le metodologie e gli strumenti atti alla sua gestione attraverso un approccio basato sull'innovazione culturale, organizzativa e tecnologica; tale approccio è finalizzato allo sviluppo di capacità e competenze in grado di aumentare la competitività dell'impresa, ottimizzandone i processi, accorciandone il “time to market” e orientandone strutture in modo più funzionale al business”;* (Sorge, 2000)
- *“Il Knowledge Management riguarda quei processi organizzativi che uniscono la capacità di combinare i dati ed elaborare le informazioni, con la creatività e la capacità di innovare degli esseri umani”;* (Malhotra, 1998)

Coniugando le tre definizioni precedenti potremo univocamente descrivere il Knowledge Management come:

“un insieme di metodologie e processi volti a gestire la conoscenza per renderla disponibile all'impresa, da tutelare e gestire, basata su processi, persone e tecnologia “

- Persone:

in quanto possessori della conoscenza, sono l'elemento più importante e più delicato che il knowledge management si trova a gestire.

Le persone sono detentrici di gran parte della conoscenza all'interno dell'azienda, soprattutto di quella "tacita", quella che non può essere gestita e trasferita nell'organizzazione. Per questo deve essere esplicitata mediante una buona gestione del knowledge management.

- Processi:

“il Knowledge Management è sottoposto alla tensione di due domande contraddittorie.

Da una parte esso si adopera per rendere la conoscenza facilmente accessibile; dall'altra, spesso il valore stesso della conoscenza è incomprensibile al di fuori del particolare contesto in cui si è sviluppato. Occorre, perciò, comprendere quali processi di diffusione dell'informazione e quali di business siano abbastanza flessibili da raggiungere un trade-off tra queste due tendenze.”

(https://www.itconsult.it/knowledge-box/white-paper/PDFitc_WP_Formazione_Cap2.pdf)

- Tecnologia:

È l'elemento che favorisce la condivisione e il trasferimento delle informazioni, si può considerare come un elemento di supporto del knowledge management, oggi diventato fondamentale soprattutto per quelle organizzazioni che si articolano in più sedi geograficamente distanti tra di loro, ad esempio a causa della recente tendenza all'outsourcing.

Il Knowledge Management perciò non è da intendersi, come spesso viene fatto, come un insieme di tecnologie (banche dati, sistemi informativi, ecc), che aiutano a trasferire la conoscenza. Infatti, in questo modo, viene attribuita importanza solo alla struttura tecnologica dell'organizzazione piuttosto che alle reali pratiche e strategie che un'organizzazione deve attuare per creare e condividere conoscenza al suo interno.

Concludendo, si può affermare che per KM s'intende “il sistema di metodologie e processi finalizzati a gestire in modo ottimale le conoscenze aziendali “critiche”; quelle conoscenze che sono necessarie per conseguire specifici obiettivi di miglioramento di singole attività nel breve

termine o per sostenere il vantaggio competitivo dell'azienda nel medio-lungo termine. Rientrano nel KM le metodologie e i processi che consentono di gestire in modo strutturato l'intero ciclo della conoscenza" (Minghetti, 2000):

- identificazione;
- acquisizione;
- diffusione;
- conservazione;
- valorizzazione;
- integrazione e aggiornamento, a livello sia di singoli processi/unità organizzative e famiglie professionali, sia d'azienda.

In definitiva "il KM ha per obiettivo il miglioramento di efficienza, efficacia e qualità d'innovazione dei processi aziendali attraverso la sistematica, strutturata e continua trasformazione dell'informazione (insieme di fatti e dati organizzati in modo da descrivere una particolare situazione o condizione) in conoscenza (insieme di fatti, verità e credi, prospettive e concetti, giudizi e aspettative, metodologie e know-how accumulati, integrati e detenuti da un'impresa lungo un adeguato arco temporale e disponibili per applicazioni operative di business nella gestione di specifiche situazioni e problemi dell'attività giornaliera)". (https://www.itconsult.it/knowledge-box/whitepaper/PDF/itc_WP_Formazione_Cap2.pdf)

3.2 Vantaggi che L'ICT ha apportato al KM

L'obiettivo del knowledge management è quello di generare e diffondere conoscenza all'interno dell'azienda, in modo da supportarne gli obiettivi strategici e incrementarne il patrimonio di risorse e competenze.

Tutto ciò comporta notevoli vantaggi in termini competitivi tra i quali: la capacità di innovazione, la reattività ai cambiamenti, miglioramento dell'efficienza, ecc.

I supporti informatici moderni possono semplificare molti elementi che stanno alla base del knowledge management, dalla classificazione e mappatura delle conoscenze alla loro reperibilità e al loro aggiornamento, è poi fondamentale per la creazione di comunità virtuali altrimenti non attuabile.

I vantaggi dell'ICT si notano soprattutto quando l'organizzazione è: multi-sede, dispersa geograficamente, di dimensioni medio-grandi e, infine, caratterizzata da un elevato livello di complessità nei suoi processi.

I sistemi di supporto informatico al KM sono composti principalmente dai seguenti elementi:

- infrastruttura di comunicazione (intranet);
- strumenti che permettono l'interazione on line (chat, blog);
- repository dove si raccoglie, organizza e diffonde la conoscenza esplicita formalizzata;
- knowledge manager che faciliti l'interazione tra sistema di supporto informatico e gli individui;
- sistemi per rappresentare e formalizzare il sapere implicito.

Da questo si evince che un sistema di supporto informatico è indispensabile ormai per implementare un efficace sistema di knowledge management, anche se non è sufficiente. Quest'ultimo infatti dovrebbe essere affiancato da una cultura aziendale che promuova la creazione e condivisione della conoscenza.

Oltretutto Il sistema di ICT deve essere coerente con l'organizzazione, non esiste un sistema di supporto informatico perfetto se non contestualizzato nell'organizzazione nel quale sarà adoperato.

Alcuni strumenti ICT alla base del KM sono: reti come internet, extranet, intranet; repository per immagazzinare dati; software che aiutano il processo decisionale e automatizzano vari processi aziendali come l'OLAP.

Nel prossimo paragrafo vedremo alcuni dei principali strumenti informatici utilizzati.

3.3 Principali strumenti ICT a supporto del Knowledge Management

Alcuni dei principali strumenti informatici utilizzati all'interno delle organizzazioni sono:

- *Repository*: fondamentali per depositare e distribuire la conoscenza. Sono sostanzialmente dei depositi on-line e su computer disponibili, per tutti i dipendenti, di informazioni inerenti all'organizzazione nel suo complesso, ai suoi prodotti, ai suoi servizi, ai suoi clienti, ai suoi fornitori e ai suoi business processes.
- *Portale Aziendale*: grazie alla diffusione delle reti Internet ed extranet, il portale aziendale diventa lo strumento principale per le organizzazioni per interfacciarsi con i propri clienti e fornitori e migliorare il rapporto con questi grazie al continuo scambio di informazioni. Nelle versioni più recenti di portali aziendali gli utenti possono inserire documenti e altri oggetti informatici senza necessità di convertirli in altro formato, consentendo così una comunicazione bidirezionale tramite il portale.
- *Intranet*: Tra gli strumenti tecnologici utilizzati troviamo l'Intranet, la rete dati aziendale, ovvero una rete di calcolatori interna all'azienda che svolge la funzione di strumento di comunicazione e scambio informativo, connettendo i computer di tutte le unità organizzative dell'azienda anche se dispersi tra stabilimenti produttivi e filiali. E' un sistema basato sulla tecnologia dei sistemi Internet ma, a differenza di questi, ha una diffusione geografica volutamente limitata all'interno dei confini aziendali.

Permette la circolazione in modo diffuso e veloce dell'informazione aziendale rendendola così maggiormente gestibile, fruibile e affidabile. Standardizzando le informazioni in modo da ottenere un più facile accesso e condividendole su una piattaforma comune, queste assumono un valore maggiore (Bava, 2003).

Può tuttavia essere molto difficile trovare le informazioni volute quando non si è a conoscenza della loro localizzazione. Questo inconveniente può essere risolto con una riorganizzazione della rete Intranet trasformandola in portale aziendale che permette l'accesso a qualsiasi tipo di informazione aziendale. Così le più evolute Intranet dispongono di alcune funzionalità standard che agevolano la circolazione di conoscenze entro l'organizzazione e sono attivabili tramite accesso a un portale aziendale.

- *Extranet*: quando si hanno comunicazioni limitate, controllate e sicure tra una rete Intranet aziendale e partners aziendali identificati, si utilizza una rete Extranet. Sempre basata sulla tecnologia Internet, consente in questo caso una sicura e tempestiva condivisione di informazioni riservate coi propri partners, fornitori e agenti.

Notevoli sono i benefici che può apportare, soprattutto in termini di efficienza degli assetti organizzativi e della logistica di aziende caratterizzate da una struttura delocalizzata e da significativi, e sistematici, flussi informativi nei confronti di altre aziende della filiera produttiva.

- *Data warehouse*: Letteralmente significa “magazzino di dati” e ha la funzione di integrazione di dati provenienti da fonti (database) diverse per generare da tali dati, già esistenti ma scollegati, nuove aggregazioni secondo diversi punti di vista.

Va utilizzato in applicazioni di supporto ai processi di decision making e consente anche l’aggiornamento e l’aggiunta di conoscenza da parte degli utenti che normalmente non interagiscono col sistema di KM.

Le caratteristiche del data warehouse sono di seguito illustrate:

- è orientato alle informazioni e non ad un problema specifico. Quindi, se nei sistemi applicativi di base ci si concentra sulle applicazioni, qui l’attenzione è posta sugli oggetti, sulle informazioni, sui fatti e non sulla loro manipolazione.
- è integrato: per eliminare le molte incongruenze tra i vari sistemi informatici di base, che usano diverse piattaforme tecnologiche, Database Management System (DBMS), codifiche, convenzioni sui nomi, attributi fisici ecc, i dati vengono inseriti in una data warehouse.
- non è volatile: contrariamente ai sistemi applicativi di base in cui i dati vengono aggiornati continuamente, qui sono effettuate fotografie successive della realtà, raccolte a cadenze prefissate. Si memorizzano così anche le modifiche, che altrimenti sarebbero eliminate.
- l’orizzonte temporale dei dati è molto superiore rispetto ai sistemi di base, contenendo la storia degli stessi (varianza nel tempo).

Quindi non è un prodotto standard ma un insieme di processi operanti su dati già esistenti, estratti da diverse fonti, che vengono poi integrati in un insieme omogeneo, memorizzati in un nuovo database costruito ad hoc e, infine, aggregati in sottosistemi specifici e producono report per l'utente finale.

- *Document management*: sono sistemi di gestione di documenti in uno dei formati elettronici disponibili. Consiste in un sistema informatico che permette la gestione di un documento, dalla creazione (documento interno) o dall'acquisizione (documento esterno) alle modifiche, all'utilizzo, alle copie effettuate e all'archiviazione.

Un sistema di KM deve gestire le informazioni documentali e permetterne l'accessibilità attraverso la memorizzazione di tutti i documenti in una repository unica, deve consentire la fruibilità mediante un accesso personalizzato alle informazioni con potenti funzioni di ricerca, deve garantire la sicurezza grazie ai controlli sull'accesso e deve assicurare la qualità attraverso la gestione delle versioni.

Per una gestione efficiente e affidabile dei documenti deve inoltre essere gestito il ciclo di vita dello stesso documento.

È ormai evidente l'importanza strategica che hanno assunto i documenti nell'era digitale. Altrettanto evidente è l'inadeguatezza dello sforzo di ogni organizzazione per gestirli al meglio. I motivi sono diversi, in particolare è stato stabilito un certo *modus operandi* che non si pensa di dover mettere in discussione.

3.3.1 Opzioni di soluzione basate su cloud

Negli ultimi due decenni sono state introdotte innovazioni nell'IT facilitandone la sua adozione, soprattutto se si considera che questo ha portato ad una notevole scalabilità nei prezzi ed una innovazione sempre più spinta, in particolare per la gestione della conoscenza di una quantità crescente di dati di dimensioni variabili. In particolare, il cloud computing e le sue soluzioni sono diventate uno dei paradigmi che soddisfa questi requisiti, supportando l'implementazione delle strategie di gestione della conoscenza. Il cloud computing è una modalità che utilizza i software e hardware virtuali (anziché fisici) forniti (da centri dati di proprietà e gestiti da fornitori di cloud e / o utenti finali) in remoto tramite reti pubbliche (ad es. Internet), reti private

o un mix (ad es. ibrido) delle due modalità di consegna in base alle esigenze e ai requisiti degli utenti.

I servizi ICT forniti includono:

- programmi per computer relativi al business (software as a service, SaaS);
- capacità di elaborazione veloci e quasi illimitate;
- strutture di archiviazione di grandi dimensioni e quasi illimitate (infrastruttura come servizio, IaaS);
- strumenti di sviluppo e opzioni di hosting per i clienti che preferiscono creare e gestire le proprie applicazioni Web (piattaforma come servizio, PaaS). (Sultan 2012, 161).

I servizi di cloud computing ottenibili dai fornitori di cloud tramite i loro data center (cloud pubblici) e gli utenti finali (ovvero le organizzazioni client) utilizzando software cloud installati sui propri data center (cloud privati) o installati sui propri centri di dati o di altri fornitori di cloud (nuvole ibride).

Considerato il potenziale offerto dal cloud computing sotto forma di servizi ICT in grado di soddisfare i requisiti di diverse organizzazioni, diverse importanti società IT e altre organizzazioni hanno iniziato a fornire soluzioni basate su cloud che possono essere personalizzate per implementare sistemi di gestione della conoscenza e relativi attività, in base alle specifiche esigenze delle organizzazioni.

Ad esempio, Microsoft, Salesforce e Atlassian hanno commercializzato le proprie soluzioni di knowledge management sfruttando le tecnologie definite in precedenza.

Inoltre, esistono soluzioni basate su open source come Moodle, Drupal, MediaWiki e WordPress (gestione dei media potenti) in grado di fornire funzionalità sufficienti per l'implementazione di un sistema di gestione delle conoscenze personalizzato basato sulla tecnologia Web 2.0.

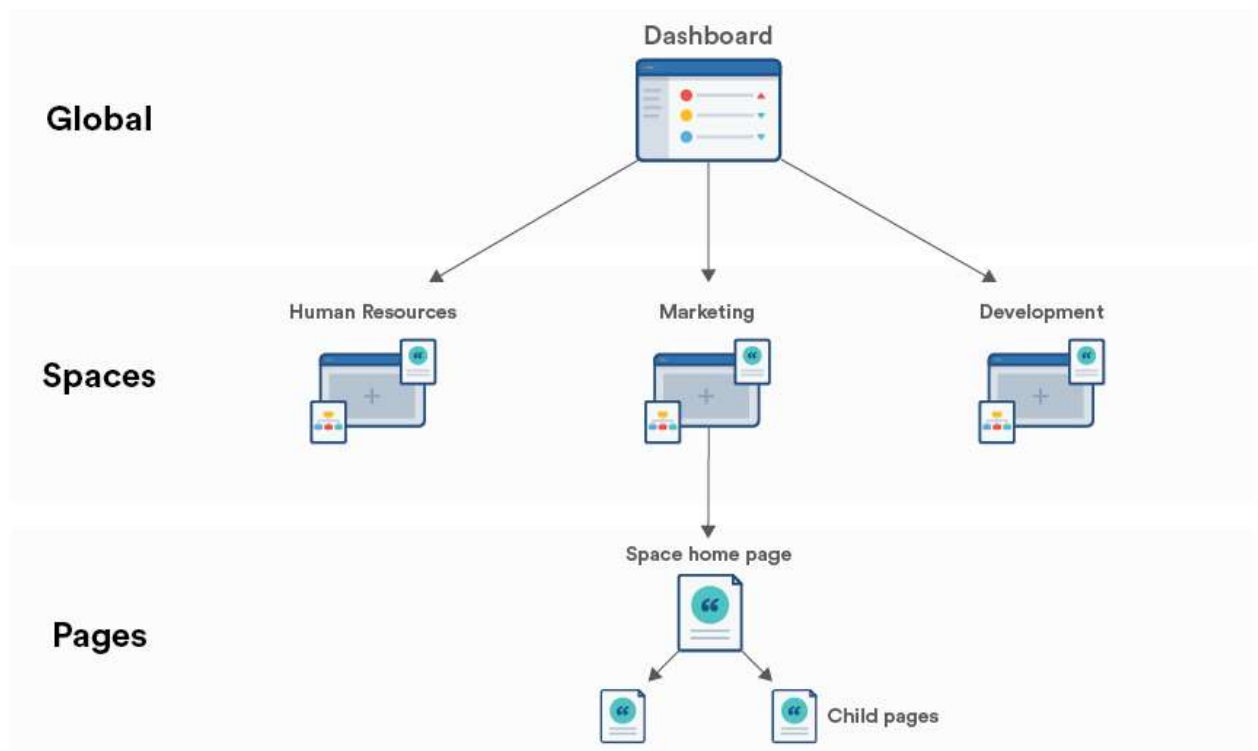
Confluence

Per favorire l'integrazione e rimanere fedele allo standard utilizzato in azienda, la scelta è ricaduta su Confluence. Infatti la gestione delle richieste e dei progetti in ambito IT all'interno dell'azienda, avviene attualmente mediante un altro prodotto di Atlassian (Jira).

Confluence e Jira sono stati progettati per integrarsi a vicenda, ad esempio offrendo agli utenti Confluence la possibilità di visualizzare, interagire e fare riferimento ai problemi di Jira da una pagina wiki.

Confluence promuove la collaborazione, la raccolta delle specifiche funzionali, durante la discussione del progetto e Jira per la visualizzazione dei problemi o la creazione di nuovi all'interno di uno stesso ambiente cambiando unicamente ed in modo immediato l'applicativo di cui si necessita.

Nel dettaglio, Confluence è uno strumento utilizzato per aiutare i team a collaborare e condividere le conoscenze in modo efficiente, organizzando i contenuti attraverso la creazione di spazi, pagine e blog che possono essere commentati e modificati da tutti i membri del team facilmente grazie alle funzionalità di condivisione. Può essere installato sul desktop o accessibile dal cloud, rendendolo un'ottima soluzione per tutti i tipi di team, inclusi dipendenti remoti e utenti mobili.



La visualizzazione di tutti i tipi di contenuti è possibile grazie al rich text editor e agli allegati.

Le autorizzazioni e le restrizioni consentono di controllare la visibilità dei contenuti a livello di spazio o di pagina in modo che solo le persone giuste abbiano accesso alle informazioni permesse.

Applicativo progettato anche per prevenire duplicati e visualizzare solo le ultime informazioni aggiornate, mediante la tracciabilità delle diverse versioni dei documenti editate.

Sebbene Confluence possa essere adattato a qualsiasi tipo di esigenza, sono stati documentati alcuni casi d'uso specifici, come l'utilizzo di Confluence come Knowledge Base o come intranet.

Sarà possibile creare una roadmap, creare note contenenti liste di controllo, creare post di blog e centralizzare tutto in un unico posto.

L'organizzazione dei contenuti potrà essere facilitata, e standardizzata, grazie all'utilizzo di modelli per l'esperienza utente unificata, etichette per la gestione dei contenuti e la possibilità di creare spazi diversi per diversi tipi di documentazione, come ad esempio spazi personali forniti agli utenti in cui potranno trovare i loro compiti. (<https://confluence.atlassian.com>)

L'applicativo permette anche funzionalità di amministrazione avanzate come la gestione degli utenti, la selezione della lingua, l'integrazione con applicazioni esterne e la gestione dei plugin e le funzioni di gestione dell'istruzione.

In particolare, Confluence offre due soluzioni: cloud o server. Nel primo caso non sarà necessaria una installazione e configurazione dell'applicativo, direttamente gestito in remoto da Atlassian, specificamente mirata per le piccole e medie imprese.

Nel caso specifico aziendale si è deciso di adottare la soluzione server grazie all'architettura informatica e il dipartimento dedicato presente all'interno dell'azienda.

Capitolo 4: Caso Di Studio

Nel prossimo paragrafo si descriverà nel dettaglio come il progetto è stato gestito nella realtà aziendale attuale e come si è ipotizzato di gestirlo mediante l'introduzione della metodologia Agile e del knowledge management tramite il tool "Confluence".

Il progetto presenta dei rischi legati a diversi fattori inerenti il contesto, i fornitori e le diverse richieste di implementazione.

I maggiori rischi che sarà possibile riconoscere sono:

- Rischio di lingua: non sempre considerato nella gestione dei progetti e che, spesso, rappresenta una delle maggiori difficoltà all'interno. Difatti, considerato il coinvolgimento di fornitori esterni (Equifax Messico) e della sede legale della banca (Madrid), è stata individuata come lingua ufficiale del progetto lo spagnolo.
- Rischi finanziari: legati alla natura del business coinvolto nel progetto. La trattazione di pratiche a recupero interessano migliaia di utenti debitori, pertanto, l'errata realizzazione ed implementazione del nuovo applicativo potrebbe portare ad elevate perdite economiche.
- Rischio di fuso orario: la sede operativa del fornitore è Città del Messico

La gestione del progetto è avvenuta sfruttando il sistema di trouble ticketing presente in azienda (Jira), facilitando il tracciamento delle richieste e del loro stato, in modo conforme agli standard strutturali ed organizzativi aziendali descritti nel prossimo paragrafo.

4.1 Situazione “AS IS”:

Nella situazione attuale la gestione del progetto “Workflow CBU” è avvenuta secondo la metodologia di Waterfall project Management.

Il progetto è stato eseguito, come da definizione, secondo diverse fasi raggruppate in macroaree suddivise per competenza, individuate in: Pre-Kick Off, Kick Off, Implementazione dell’infrastruttura, design, strategia e data governance; esplicitate nel gantt seguente:

	Nome Attività	Status	Inizio	Fine	Timing	% Completamento	Ownership	Referente
1	Pre-Kick Off					63%		
1.6.1	Anàlisis del procesos as-is	Done	mer 22/11/17	gio 23/11/17		100%	BPG	
2	Kick Off					21%		
3	Implementación de Infraestructura					0%		
4	Diseño					0%		
5	Estrategias					0%		
6	Data Governance					0%		

La fase iniziale di pre-kick-off meeting è stata affidata al PMO individuato, nella quale ne ha definito clausole, date di scadenze, ownership, risorse, responsabilità e tutto ciò che concerne la realizzazione del project charter e la gestione delle fasi di “Avvio” e pianificazione.

Lo stesso PMO ha supervisionato e monitorato le successive fasi, invece assegnate e realizzate dalle diverse aree e direzioni interne della banca secondo la loro competenza.

Nel dettaglio il lavoro di tesi osserverà come può avvenire la gestione agile di una di queste macrofasi del progetto, che vede maggiormente coinvolta la parte IT, che sarà trattata come un progetto costituita da attività, risorse, scadenze.

La gestione di questa fase del progetto che, per osservanza procedurale, è avvenuta mediante il tool di trouble ticketing successivamente descritto, attraverso la creazione e risoluzione di diverse richieste, classificate per tipologia, priorità, ambito e necessità differenti.

In merito alla gestione delle risorse, attualmente, si considera unicamente la disponibilità totale giornaliera di ognuna, impiegata in modo trasversale su più progetti, non definendo una capacità predefinita per ogni progetto e per ogni individuo. Ad oggi, per quanto precedentemente descritto, non è possibile avere una pianificazione puntuale delle attività assegnabili alle risorse solo in modo itinere e secondo la loro disponibilità.

In particolare nella descrizione della situazione attuale sarà di fondamentale importanza la definizione dei ruoli e funzioni presenti nella direzione IT e delle diverse tipologie delle richieste segnalate dagli utenti esterni/interni.

4.1.1 Definizioni

I sistemi di trouble ticketing

I sistemi di trouble ticketing hanno come scopo quello di gestire i problemi che vengono sollevati dagli utenti, in modo da ottimizzare i processi aziendali e la produttività complessiva. Si espone il funzionamento di tali sistemi nella sezione seguente.

In un sistema tradizionale, un generico cliente segnala un problema, generalmente tramite posta elettronica e, nel caso non riceva alcuna risposta da nessun operatore, provvede al sollecito con nuovi messaggi.

Essendo tali messaggi presenti in una mailing list, i differenti operatori non sono a conoscenza né se il problema è stato risolto, né tantomeno chi è colui che lo sta risolvendo.

Con i sistemi di trouble ticketing, detti anche di trouble report, invece, per ogni richiesta fatta da un cliente viene generato un ticket (richiesta/segnalazione), che viene salvato in una base di dati. In questo modo tutti gli operatori in grado di risolvere il problema sono a conoscenza, non solo della richiesta effettuata dal cliente, ma anche della persona che se ne sta occupando.

È possibile, inoltre, che più persone in sequenza risolvano il problema sollevato, ma l'utente finale è a conoscenza solo dello stato in cui si trova un ticket in un istante di tempo e della eventuale risoluzione o meno del problema e non della sequenza completa delle comunicazioni realizzate tra i differenti operatori.

Quest'ultimi possono essere organizzati in gruppi di lavoro, in modo da collaborare tra loro per poter risolvere il problema sollevato da un utente o risolverlo individualmente.

Per ogni ticket, a seconda del problema segnalato, l'operatore incaricato vi assegna una priorità, in base alla quale, sono stabiliti dei tempi di risoluzione che devono, o dovrebbero, essere rispettati.

Il ticket viene considerato chiuso quando viene inviata la risposta al cliente.

Per quanto riguarda l'azienda committente, il sistema di trouble ticketing utilizzato è il sistema Jira di Atlassian.

Ruoli di progetto

Il progetto vede in prima linea diversi attori, interni ed esterni al contesto aziendale.

Nel dettaglio, i principali attori coinvolti nel progetto che potremo identificare sono:

- Fornitore esterno: “Equifax”, leader mondiale tra le consumer credit reporting agency con sede legale in Atlanta, che presterà assistenza nell’integrazione dell’applicativo all’interno della banca e della manutenzione, post rilascio in produzione, in caso di malfunzionamenti.
- Struttura dell’IT: struttura della banca che coopererà con i fornitori esterni, in maniera integrata, occupandosi delle modifiche interne da dover effettuare per l’implementazione dell’applicativo. La struttura consta di cinque ruoli principali:

1. Demand Manager

Tale ruolo rappresenta il punto di contatto del business con l’IT e garantisce una gestione univoca, omogenea e strutturata attraverso: l’indirizzamento corretto delle richieste (anche secondo il livello di approvazione richiesto); assicurando il coinvolgimento di tutti gli attori/unità organizzative impattate; presidiando la fase di UAT e raccogliendo tutte le approvazioni necessarie del Business.

2. Programmatore/Sviluppatore

Il ruolo di Programmatore/Sviluppatore è interno a Santander e si occupa dell’implementazione dei requisiti ottenuti attraverso la risoluzione delle segnalazioni e richieste assegnate.

3. Responsabili Organizzativi

Supervisionano e coordinano tutte le attività e risorse dell’organizzazione di cui sono responsabili.

Le principali mansioni quotidiane che quest’ultimi sono tenuti a svolgere sono:

- Formare, guidare, supervisionare e disciplinare lo staff.
- Garantire l’efficienza generale delle operazioni giornaliere, monitorando giorno per giorno il flusso di lavoro, e delegando coerentemente i compiti tra i membri dello staff.

- Monitorare e tracciare le prestazioni dei dipendenti, il budget progettuale e le rispettive scadenze.
- Preparare presentazioni, proposte e resoconti.

4. DWH

Struttura che si occupa della raccolta di dati "integrata, orientata al soggetto, variabile nel tempo e non volatile" di supporto ai processi decisionali. L'integrazione dei dati costituisce la principale caratteristica distintiva del DW rispetto ad altri sistemi di supporto alle decisioni.

- Business: è identificato con CBU, l'unità della banca coinvolta in primis nel progetto e che attualmente si occupa della gestione del recupero crediti così che possa beneficiare dei risultati ottenuti con lo stesso.

Tipologia Richieste

Le richieste, attualmente gestite all'interno del tool, vengono classificate in base alle definizioni dell'ITIL nelle seguenti tre tipologie:

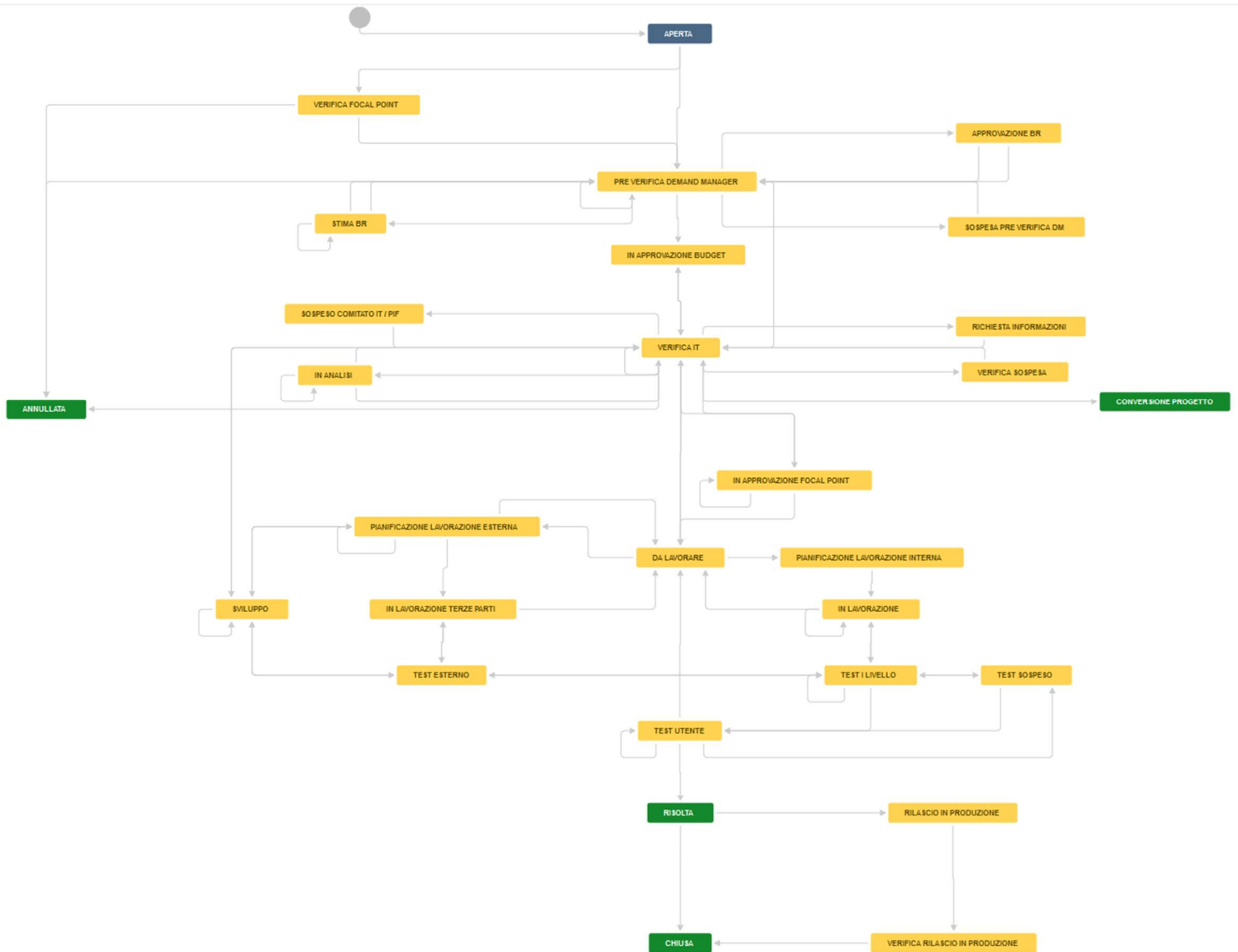
- **Incident Request:** richieste legate a qualsiasi evento che pregiudica l'operatività standard di un applicativo e che causa, o può causare, un'interruzione e una riduzione della qualità (ad esempio: blocco funzionalità, campo calcolato non correttamente, anomalie, ecc.)
- **Service Request:** richieste che fanno riferimento a modifiche di funzionalità esistenti o a servizi standard IT legati a hardware (PC, stampanti, server, ecc.), gestione utenze (creazione utenze di sistema, di VPN, modifica punti di menù o profili, ecc.), software (installazione Visio, richieste licenze, ecc.) o supporto (richiesta di informazioni o estrazioni, ricerca documentale, ecc.)
- **Change Request:** richieste che fanno riferimento alla creazione di nuove funzionalità (ad esempio: creazione nuovi campi, nuova reportistica in DWH, aggiornamenti dei sistemi per le nuove normative, ecc.) e quindi che necessitano di uno sviluppo.

Ogni tipologia di richiesta segue un workflow dedicato e specifico che racchiude le azioni disponibili sulla segnalazione visualizzata e la vita della segnalazione dall'apertura alla

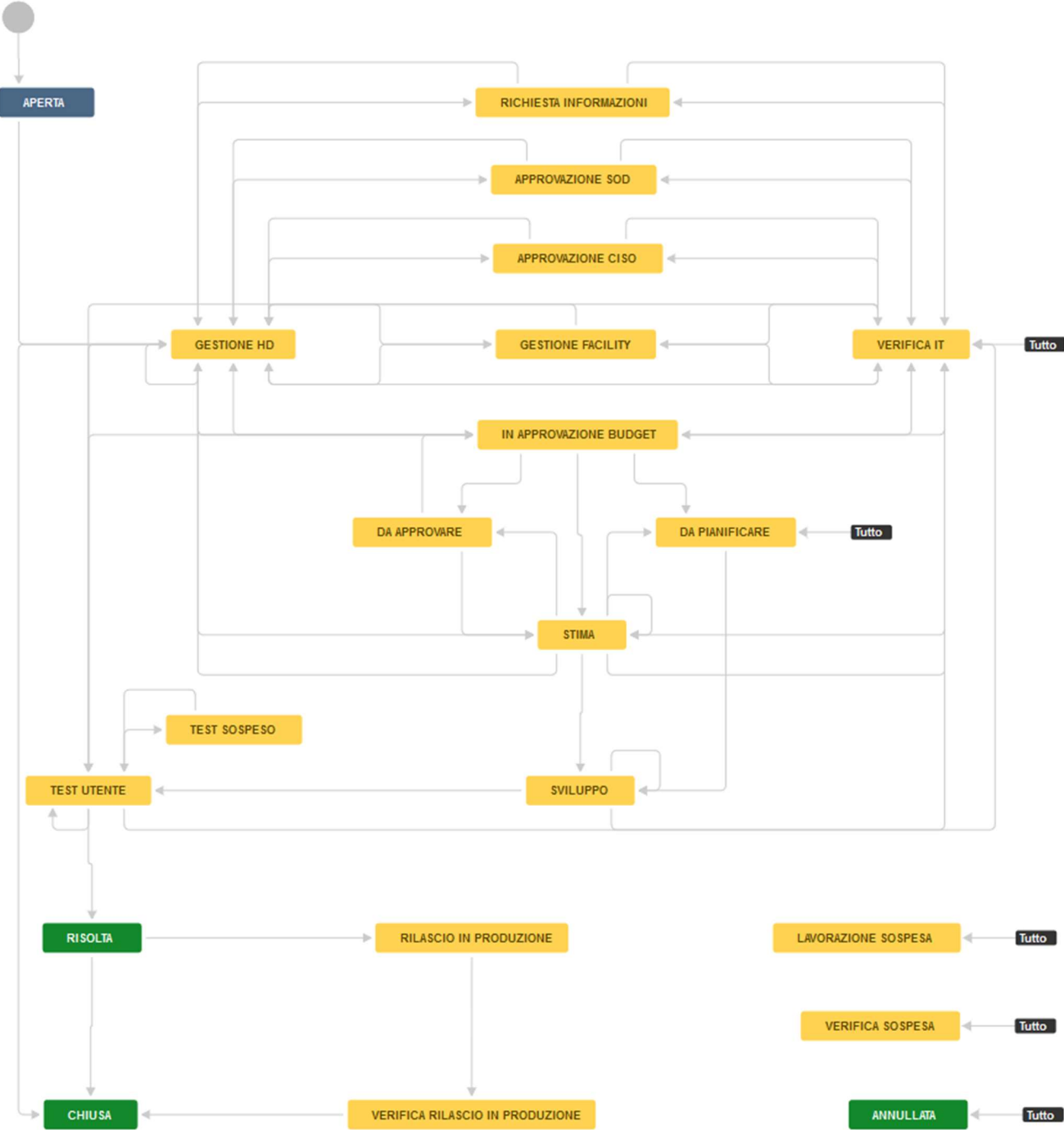
chiusura, indicando tutti i processi di autorizzazione, budgeting, sviluppo e messa in produzione delle richieste.

I workflow per ogni tipologia di richiesta sono i seguenti:

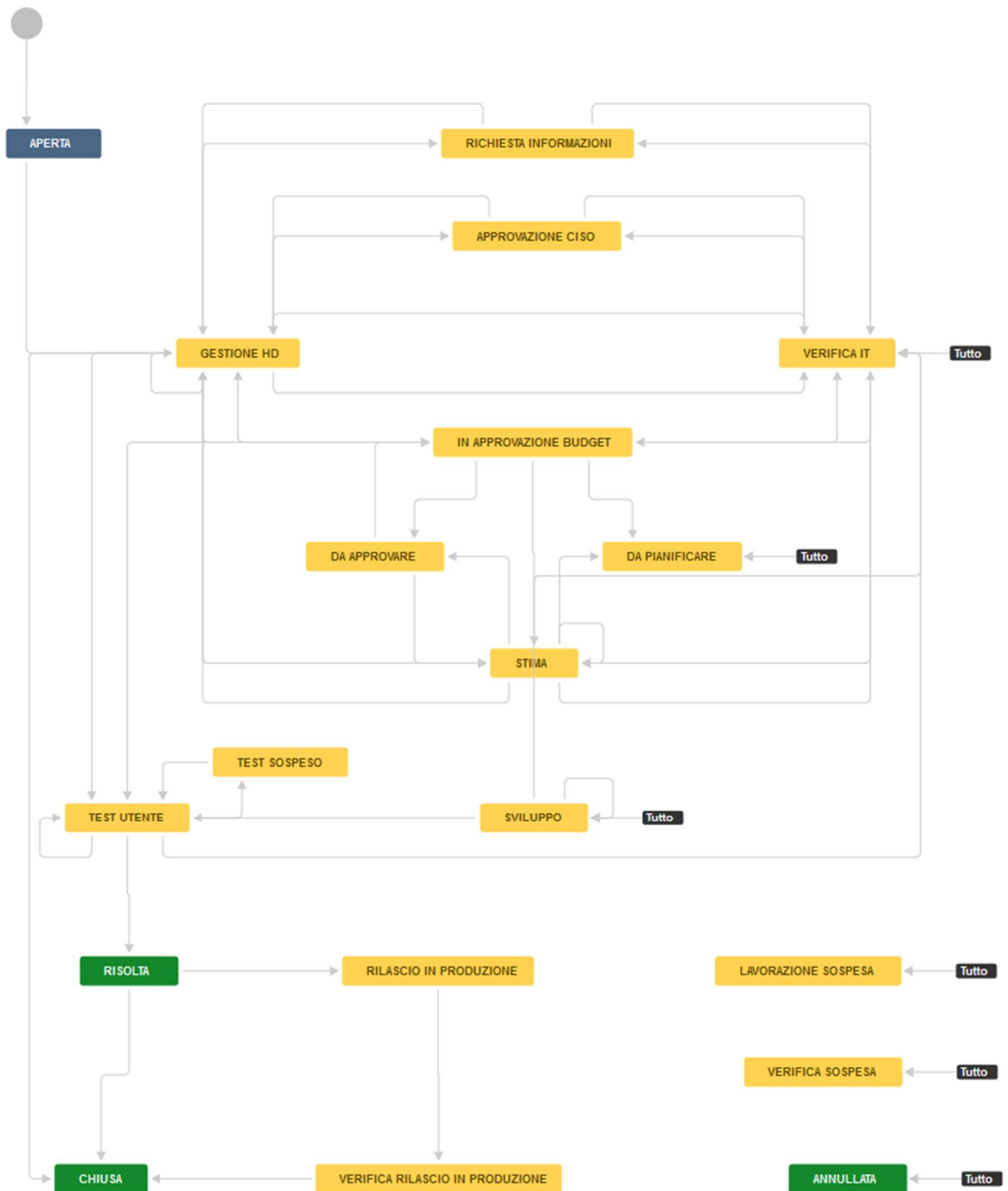
WORKFLOW CHANGE REQUEST



WORKFLOW SERVICE REQUEST



WORKFLOW INCIDENT REQUEST



4.1.2 Pianificazione delle attività

Una volta definite le diverse le diverse tipologie di richieste esistenti e i diversi stati che ognuna dovrà assumere, in questo paragrafo, potremo ora definire come avviene la pianificazione e gestione del progetto e quindi delle richieste nel sistema attuale, differenziandone, in dettaglio, i diversi approcci di gestione seguiti in funzione della tipologia di richiesta da dover gestire tra quelle prima elencate.

a) Change Management:

La tipologia di richiesta “CR” è stata utilizzata per l’implementazione dei requisiti richiesti dagli stakeholder.

Quest’ultima necessitando di un nuovo sviluppo seguirà, come da workflow precedente, un ciclo di vita più complesso e prolungato nel tempo.

Tale sviluppo dovrà essere documentato per ogni richiesta, secondo la natura prescrittiva del waterfall management, da una documentazione dettagliata, in cui analizzare i nuovi requisiti funzionali e tecnici che il richiedente necessita e che il team di sviluppo dovrà sviluppare.

La definizione dei requisiti, per ogni nuova richiesta di implementazione, attualmente, avviene mediante l’organizzazione di incontri tra i vari stakeholders, interessati allo sviluppo, il responsabile d’ufficio e i Demand, che provvederanno ad organizzare gli incontri secondo l’urgenza e le necessità.

Stakeholder: Con questo termine si fa riferimento a tutti quei soggetti portatori di interessi nei confronti dell’azienda. Essi condizionano la gestione aziendale, assicurandone, in alcuni casi, la prosperità e lo sviluppo oppure, in altri casi, facendosi portatori di minacce. Questi possono essere interni od esterni. Gli stakeholders interni si distinguono in proprietari e non proprietari: i primi sono i soci detentori di capitale di rischio; i secondi sono i dipendenti o il management non proprietario. Gli stakeholders esterni rappresentano tutti i soggetti che operano al di fuori della sfera aziendale, quali i fornitori, i clienti, i concorrenti, lo Stato.

Al termine della riunione, i membri del demand assegnati al progetto, provvedono a formalizzare tali requisiti in un’analisi chiamata “Developer Documentation Requirements (DDR)” e all’apertura delle richieste necessarie allo sviluppo degli stessi all’interno del tool aziendale.

La documentazione sarà allegata ad ogni richiesta, la quale terrà traccia di tutti i cambiamenti ed i progressi dello sviluppo, fino alla risoluzione e chiusura.

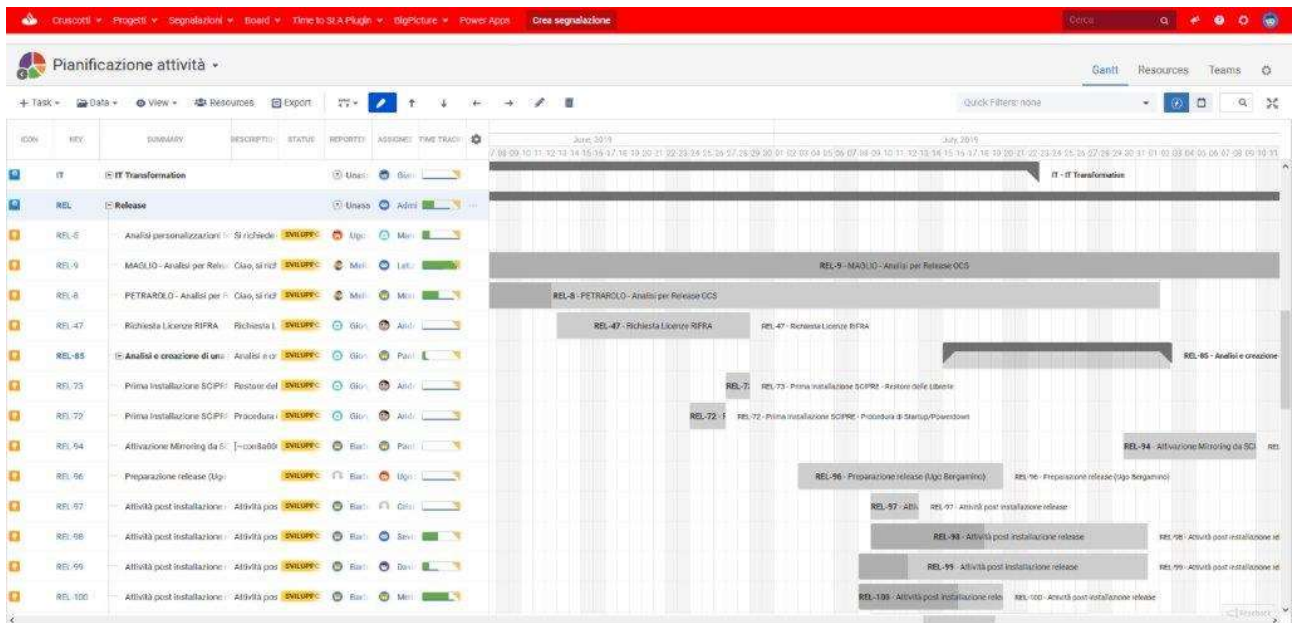
In seguito alla scrittura del documento funzionale, i membri del Team di sviluppo, dovranno visionarlo ed analizzarlo, definirne la fattibilità di realizzazione, il numero di attività tecniche necessarie da dover svolgere per l'implementazione e la stima del tempo necessario per lo sviluppo.

Le attività tecniche, definite dal team di sviluppo, passeranno nuovamente in carico ai demand che dovranno assegnarle e pianificarle sulle risorse del Delivery, oppure, se necessario suddividerle in più sotto-attività, ad esempio se complesse o se necessitano di essere assegnate a più utenti per svilupparle.

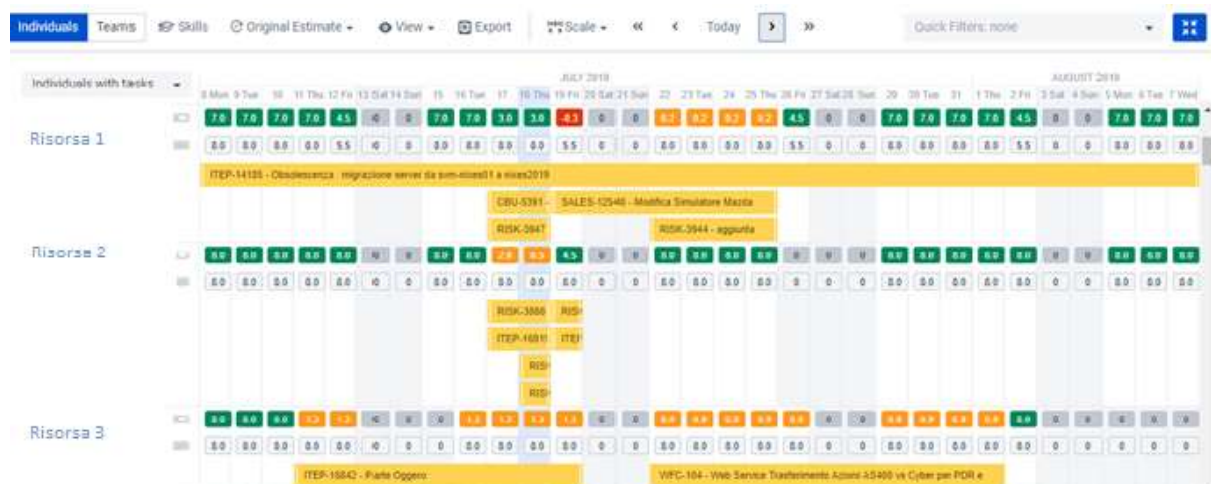


L'assegnazione delle attività che avviene, come spiegato anche precedentemente, secondo la priorità e l'ambito tecnico sulle risorse, al più presto possibile, nel loro primo slot di tempo disponibile, osservando gli eventuali vincoli funzionali e temporali tra le diverse richieste di sviluppo.

La schedulazione delle attività sulle risorse per più progetti è facilitata da Jira grazie alla creazione di un gantt che indicherà, in maniera visiva, il periodo di decorrenza delle attività già assegnate, suddivise per progetto di riferimento:



E' possibile una visualizzazione sinottica delle attività assegnate alle singole risorse, i tempi collegati, mediante l'unione tra la Resource Breakdown Structure ed il gantt:



Grazie a questo tipo di visualizzazione del gantt, gli sviluppatori potranno avere una chiara visione delle richieste che dovranno svolgere giorno per giorno ed entro quanto tempo ultimarle, e d'altra parte i Demand e i referenti organizzativi, potranno tenere facilmente traccia dell'impiego delle risorse e di eventuali sovra-allocazioni delle capacità delle stesse, evidenziate dal tool con colore rosso, potendo così rischedulare le attività su diverse risorse o in tempi diversi.

Nel dettaglio l'implementazione dell'applicativo "Cyber Financial" ha necessitato di diversi nuovi sviluppi per i quali è stato necessario aprire ventuno richieste di tipologia CR, che singolarmente hanno contribuito alla realizzazione dei requisiti funzionali richiesti dal business.

Le nuove implementazioni hanno reso necessario l'intervento di più attori nei vari ambiti applicativi del settore IT di seguito elencate:

- DWH ha dovuto occupare, in modo discontinuo, una risorsa nella creazione dei registri di sistema dai quali recuperare le informazioni che l'applicativo Cyber dovrà processare ed elaborare.
- L'IT ha dovuto fornire 2 risorse che si occupassero delle implementazioni lato web, lato AS400 e lato finanziario.
- Il fornitore esterno invece ha dovuto fornire una risorsa che si trasferisse in loco, al fine di apportare le modifiche necessarie all'applicativo, in modo da integrarlo con il sistema attuale e customizzarlo secondo le necessità ed esigenze del Business.
- Il Business ha dovuto fornire una risorsa che, identificata per comodità e tempo in un consulente esterno, deve verificare, dopo l'implementazione delle attività, la loro congruenza con le esigenze iniziali nella fase di test.

b) Service Management:

Diversamente dalla tipologia di richieste precedente, le Service requests potranno essere gestite in maniera più flessibile, infatti trattandosi di attività manutentive non compromettenti, riguardanti interventi di supporto agli sviluppi principali (Es: creazione utenza, abilitazione a cartella, o modifica di un parametro già esistente) su qualcosa di già esistente, non comporteranno grosse difficoltà o urgenze, e non necessiteranno di documentazione funzionale per la loro implementazione, ma solo della scrittura di eventuali procedure di intervento al fine di velocizzarne la risoluzione e standardizzarne le modalità di intervento.

Quindi tali richieste sono gestite, come le altre, dai demand che, successivamente ad un'analisi delle stesse da parte del team di sviluppo, dovranno inserirle all'interno della schedulazione, tenendo conto della la disponibilità delle risorse, dei vincoli funzionali, ad esempio con le richieste a cui sono di supporto, ed eventuali vincoli temporali.

A supporto del progetto è stato necessario gestire e aprire 51 richieste di tipo SR, prevalentemente associate a creazione utenze, abilitazioni e modifiche varie, necessarie all'integrazione del nuovo applicativo "CyberFinancial" ai sistemi attuali informatici e procedurali interni alla banca.

La pianificazione e l'assegnazione delle attività ai membri del team di sviluppo avviene mediante la visualizzazione delle richieste aperte, da parte degli utenti, all'interno di un bucket di problemi aperti nel tool aziendale sotto la sezione del progetto, o all'apertura ad hoc di quest'ultime secondo le necessità presentatesi durante lo sviluppo e segnalate dallo stesso team.

The screenshot displays the Jira interface for the 'Workflow CBU' project. On the left, a sidebar shows navigation options like 'WFC board', 'Backlog', 'Sprint attivi', 'Versioni', 'Report', 'Problemi', 'Componenti', and 'Documents'. The main area is titled 'Problemi aperti' and shows a list of issues. The selected issue, WFC-73, is 'blocco dello smistamento per pratiche nel perimetro di cut off aperto'. The detailed view on the right shows the following information:

- Tipi:** Change Request
- Priorità:** P5
- Stato:** VERIFICA RILASCIO L...
- Risoluzione:** Non risolta
- Etichette:** Nessuno
- Ambito/Applicativo:** Main System - AS400
- Ambiente di rilascio:** Test
- Richiesto passaggio in produzione:** SI
- Esito Test primo livello:** OK
- Esito Test Esterno:** OK
- Esito Test secondo livello:** OK
- Sia Semaforo:** SLA SUPERATO
- Prodotto:** Tutti
- Priorità Utente:** Basso
- Tipologia di Richiesta:** Funzionalità (Modifica/Creazione/Eliminazione)
- Beneficio dell'intervento:** Non deve smistare pratiche presenti in un cut off aperto
- Vincoli particolari legati alla richiesta:** Non deve smistare pratiche presenti in un cut off aperto
- Descrizione (dove):** Non deve smistare pratiche presenti in un cut off aperto

The 'Persone' section on the right lists various roles and their assignments, including 'Assegnatario: Non assegnata', 'Segnalante: Cinzia Divella', and 'Assegnazione Gruppo: Jira-Delivery'.

c) Incident Management:

La gestione degli Incident, essendo per definizione dei mal funzionamenti e problemi del sistema causanti inoperatività e disfunzioni, sono caratterizzati da una certa urgenza, che dovrà essere risolta nel più breve tempo possibile. Le richieste di questo tipo, dunque, devono essere risolte in modo pressoché immediato, quindi i Demand hanno il compito di allocarle sulle risorse di sviluppo, a prescindere da quello quest'ultime stiano facendo, cercando solo di sfruttare per prime le risorse disponibili e specializzate nell'ambito del problema verificatosi.

Tuttavia l'applicativo non ha riscontrato problemi, e gli stessi, essendo ancor in fase di sviluppo sono stati risolti tramite l'apertura di SR. Non causando inoperatività di un qualcosa ancora non operativo.

4.1.3 Problemi Riscontrati

L'organizzazione attuale di gestione dei progetti, se pur metodica e delineata da procedure, risulta essere molto complessa, non lineare e non sempre puntuale con le scadenze. Le cause sono principalmente: workflow troppo complessi, ritardi complessivi, errata gestione delle risorse, ad esempio impegnate nell'analisi della documentazione per un certo tempo che però attualmente non viene considerato e schedato nel gantt, e la gestione degli imprevisti non flessibile che costringe le risorse ad interrompere le attività di sviluppo.

In particolare la gestione Waterfall, di tipo prescrittiva, costringe alla scrittura di documenti funzionali per ogni fase e richiesta, che però non avviene sempre in modo puntuale a causa del carico di lavoro affidato alle risorse del Demand Office e all'accumulo delle richieste da implementare, che gli costringe a dover lavorare spesso in situazione di emergenza, nella quale scrivere documentazione in tempi molto brevi, spesso ripetitiva, per ogni singolo sviluppo, non dotata del livello di dettaglio e chiarezza richiesto.

Il problema della definizione e gestione della documentazione verrà risolto nella situazione "TO BE", grazie alla metodologia Agile e soprattutto grazie all'introduzione di Confluence.

Ulteriori ritardi dovuti alla suddivisione del progetto in attività di durata prevista troppo elevata, difficilmente stimabili e eseguibili completamente, da dover poi ri-suddividere, successivamente al consulto del delivery, in attività più piccole da dover assegnare alle singole risorse, necessitanti quindi di ulteriore tempo per stima, gestione e pianificazione.

In particolare, ad oggi, la definizione delle attività e la schedulazione delle stesse in un progetto spesso viene fatta a ritroso, partendo dalla data di consegna del progetto fino ad oggi. Tuttavia la pianificazione ottenuta, considerando unicamente questo vincolo, non sempre sarà concorde con risorse, tempo e elementi a disposizione in azienda, costringendo, in corso d'opera, a dover far slittare più volte la deadline e a nuove schedulazioni ripetitive delle attività sulle risorse.

Queste continue variazioni causanti ulteriori difficoltà nel monitoraggio del progetto e del suo andamento, risultando molto ostico da eseguire, poco preciso e attendibile.

Tutto ciò che porterà conseguenze su costi, qualità, oltre che su tempi e organizzazione dei progetti.

A livello informatico ed organizzativo, inoltre, alcune criticità sono dovute all'errata definizione di progetto utilizzata nel Tool, dove viene utilizzato come un tema o raccogliitore

nel quale inserire tutte le richieste, di qualsiasi tipo, tra quelle proposte, appartenenti a quel tema.

Ad esempio esistono progetti per ogni Direzione funzionale della banca (Es. Direzione CBU, Direzione IT, ecc.) o per macro-temi riferiti a servizi all'interno delle direzioni offerti dalla banca, per i quali è necessario attuare attività ricorsive, di manutenzione o aggiornamento degli stessi. Definizione contraddittoria rispetto alle definizioni riconosciute e descritte precedentemente.

4.2 Situazione “TO BE”:

In merito ai problemi definiti nella situazione attuale, l’obiettivo dell’introduzione della metodologia agile è stato quello di avviare un cambiamento strutturale nella gestione delle richieste, grazie alla quale poter avere dei workflow più semplici da gestire, una risoluzione degli imprevisti più flessibile, grazie ad un turnover delle risorse, un’implementazione dei requisiti più puntuale e organizzata.

I progetti, come nella situazione precedente, verranno gestiti attraverso la creazione e risoluzione di diverse richieste, classificate per tipologia, priorità, ambito e necessità differenti.

Tuttavia le tipologie utilizzate saranno differenti e seguiranno tutte lo stesso workflow semplificato che descriveremo più dettagliatamente nel prossimo paragrafo.

Le risorse saranno gestite in modo più puntuale, attribuendo a ciascuna una dettagliata capacity (tempo di lavoro a disposizione) per progetto su cui è allocata. In tal modo si potrà avere un maggior monitoraggio del lavoro eseguito e della percentuale di completamento dei vari progetti, grazie anche ad una pianificazione più precisa e semplificata. Infatti mediante lo Scrum proveremo ad eseguire gli sviluppi dei prodotti/servizi necessari in una serie di iterazioni di durata fissa chiamati sprint.

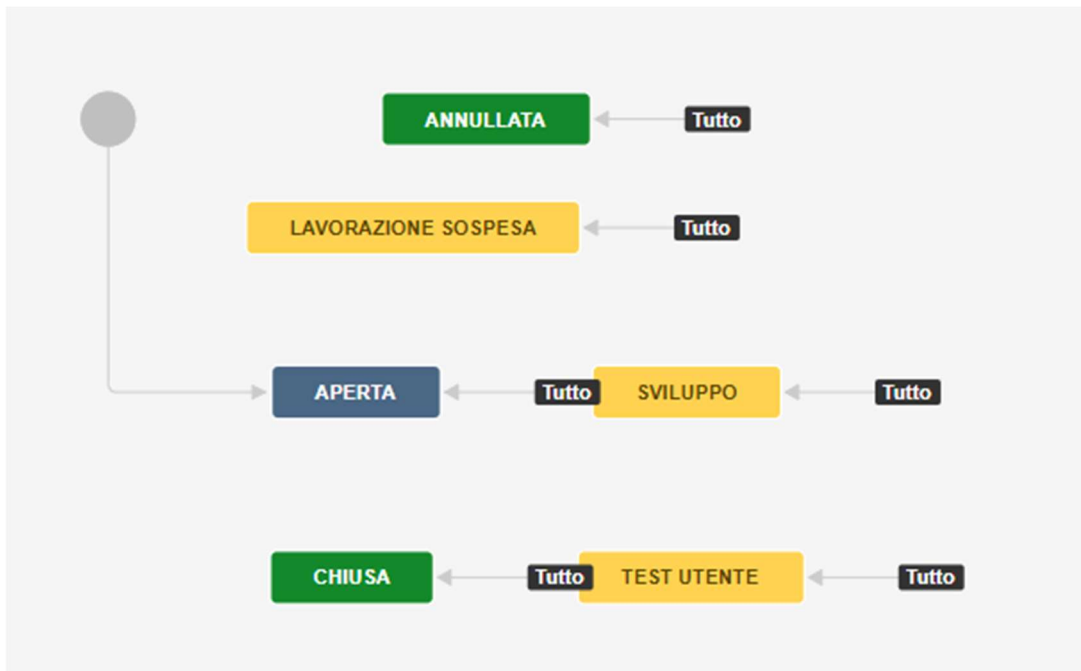
Nel dettaglio quattro pilastri costituiscono la struttura di questo framework: pianificazione rapida, standup (chiamati anche scrum quotidiani), sprint e analisi retrospettive.

Pronto all'uso, Jira Software è dotato di un set completo di strumenti agile che aiutano i team scrum a ottenere facilmente buone prestazioni in queste fasi.

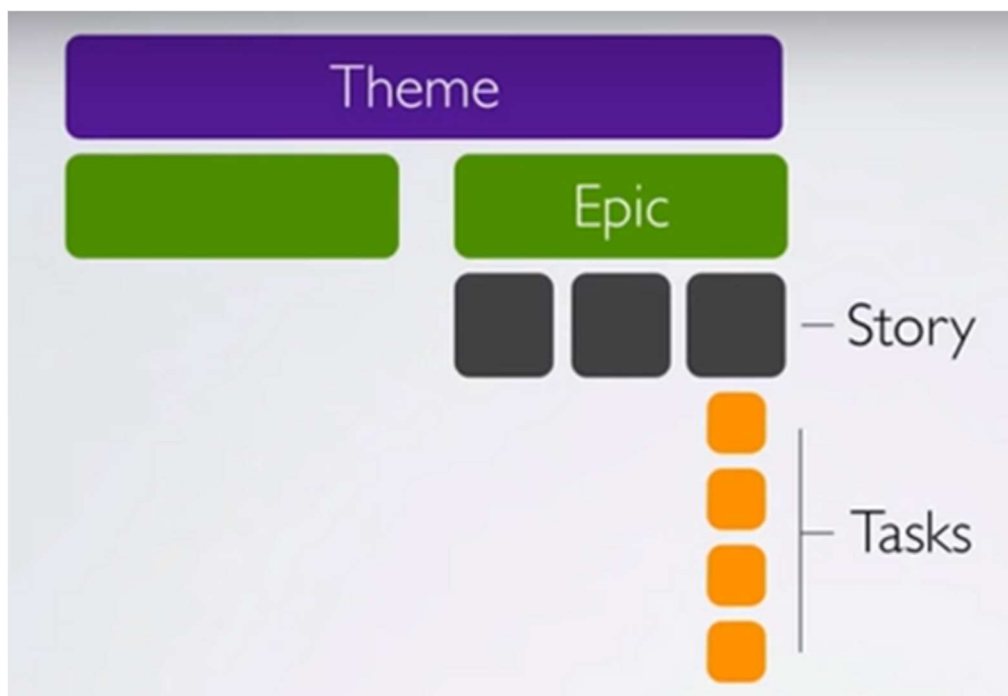
4.2.1 Definizioni

Tipologia Richieste

Grazie alla nuova metodologia verranno introdotte nuove tipologie di richieste, definite successivamente, le quali seguono tutte un workflow semplificato composto unicamente da sei stati:



Strutturate gerarchicamente in questo modo:



- **User Story:**

la descrizione ad alto livello di una funzionalità utile al raggiungimento di un obiettivo di business. Con 'storia' definiamo in maniera non dettagliata una funzionalità che un software deve avere e che dà valore al prodotto finale consegnato al cliente/utente.

Una 'storia' ci permette di descrivere ed evidenziare l'importanza e l'impatto che una funzionalità avrà nel business, aiuta a far capire – e decidere – al cliente l'utilità della funzione stessa e la sua priorità, in alcuni casi fino a decidere di non realizzarla affatto.

Il format classico di una storia è il seguente:

COME < ruolo >

VOGLIO < fare qualcosa >

COSI CHE < possa ottenere valore per il business >

- **Epic:**

una grande quantità di lavoro che può essere scomposto in un numero di storie più piccole o talvolta chiamati "Problemi" in Jira. Gli Epic comprendono spesso più team, su più progetti e possono essere monitorate su più Board.

- **Task:**

l'unità elementare da dover sviluppare in modo tecnico al fine di realizzare la necessità richiesta.

Adattamento ai Nuovi Ruoli

Nell'attuazione della nuova gestione è stato necessario riorganizzare i ruoli attualmente presenti, in modo da adattarli a quelli definiti dallo Scrum e utili alla gestione agile dei progetti.

Infatti nello Scrum sono previsti differenti ruoli:

- **Product Owner**

Il Product Owner è la figura che ha il ruolo di identificare le caratteristiche del prodotto ed assegnarne un livello di priorità.

Attraverso l'identificazione delle priorità, il product owner ha l'importante compito di stabilire l'ordine con il quale le diverse caratteristiche verranno portate in sviluppo all'interno degli sprint successivi.

Nella situazione aziendale si è pensato di far assumere tale ruolo ai membri del demand, in quanto rappresentano la figura più vicina agli stakeholder del progetto e i più informati sui requisiti da dover implementare, potendone facilmente definirne la priorità all'interno del product Backlog.

- **Il team di sviluppo**

Il team ha il duplice compito di:

- a. sviluppare le caratteristiche di prodotto individuate dal Product Owner
- b. fornire idee al Product Owner su come migliorare il prodotto
- c. Cross-funzionale, includendo tutte le competenze necessarie per il rilascio dei prodotti inseriti all'interno dello sprint backlog

Nella situazione aziendale il team di sviluppo rimarrà invariato, l'unica differenza si presenterà nella creazione di sotto team di sviluppo, formati da un numero ridotto di membri, da allocare sui diversi progetti per un tempo ben definito per progetto. Team composti da elementi multidisciplinari che potranno occuparsi di tutte le necessità richieste e che potranno cooperativamente incrementare le conoscenze e capacità personali di ogni membro, al fine di rendere l'ambiente di lavoro il più omogeneo e funzionale possibile.

- **Scrum Master**

Lo scrum master rappresenta un ruolo fondamentale per la corretta applicazione della metodologia Scrum.

Nel dettaglio, i compiti della figura in esame comprendono:

- a. Coaching
- b. Contribuire a rimuovere gli ostacoli
- c. Aiutare a risolvere i problemi
- d. Fornire Input creativo
- e. Guidare lo sviluppo delle competenze dei membri del Team

Analizzando le figure professionali attualmente presenti all'interno della struttura aziendale, per affinità, la figura che si presta maggiormente a rivestire il ruolo di scrum master è l'attuale responsabile d'ufficio, unica figura che interagisce con tutti i ruoli interessati al progetto e che quindi può in maniera agevole coordinare tutte le fasi e svolgere le attività dettagliate precedentemente.

4.2.2 Assunzioni

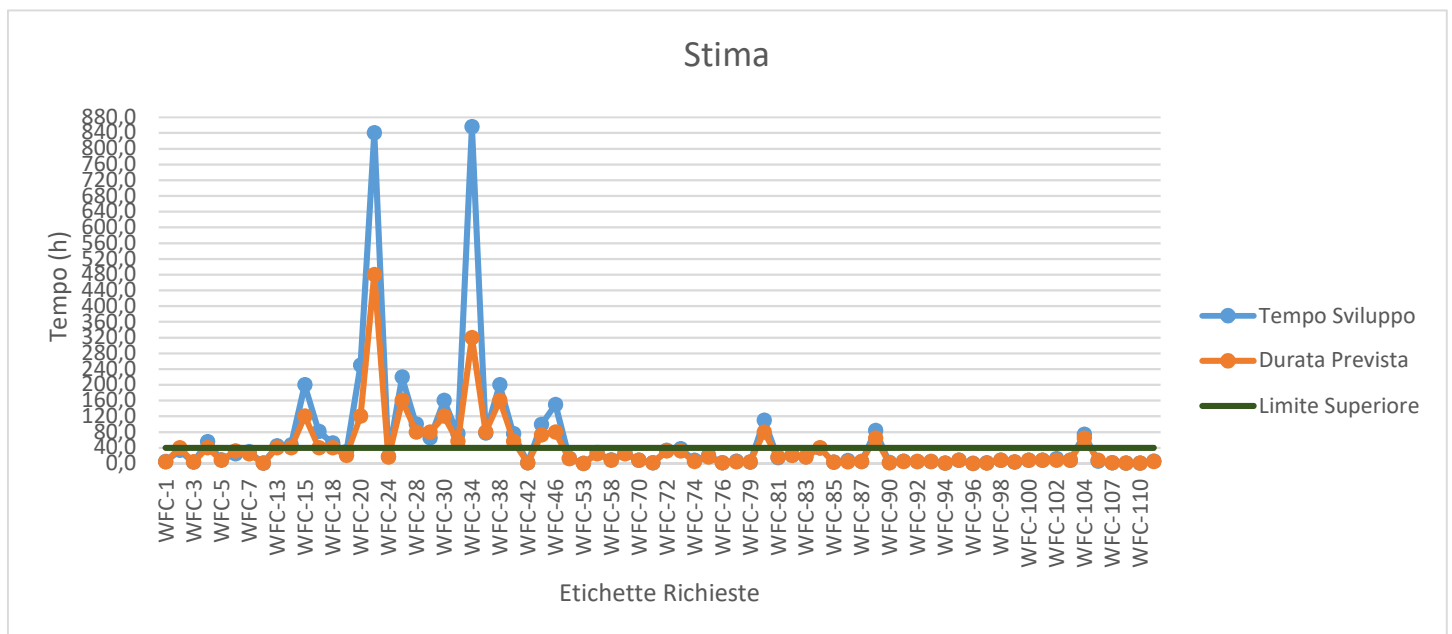
A sostegno della pianificazione delle attività ipotizzata è stato necessario definire le seguenti assunzioni, in modo tale da poter considerare la stessa veritiera e molto vicino alla realtà. Tuttavia la verifica del lavoro supposto si potrà avere unicamente nei prossimi mesi, quando l'applicazione della modalità agile definita sarà eseguito.

Nel dettaglio le assunzioni definite riguarderanno il tempo di sviluppo delle attività ed i tempi di test necessari per la verifica della attività, punto di partenza per la pianificazione e l'esecuzione degli Sprint.

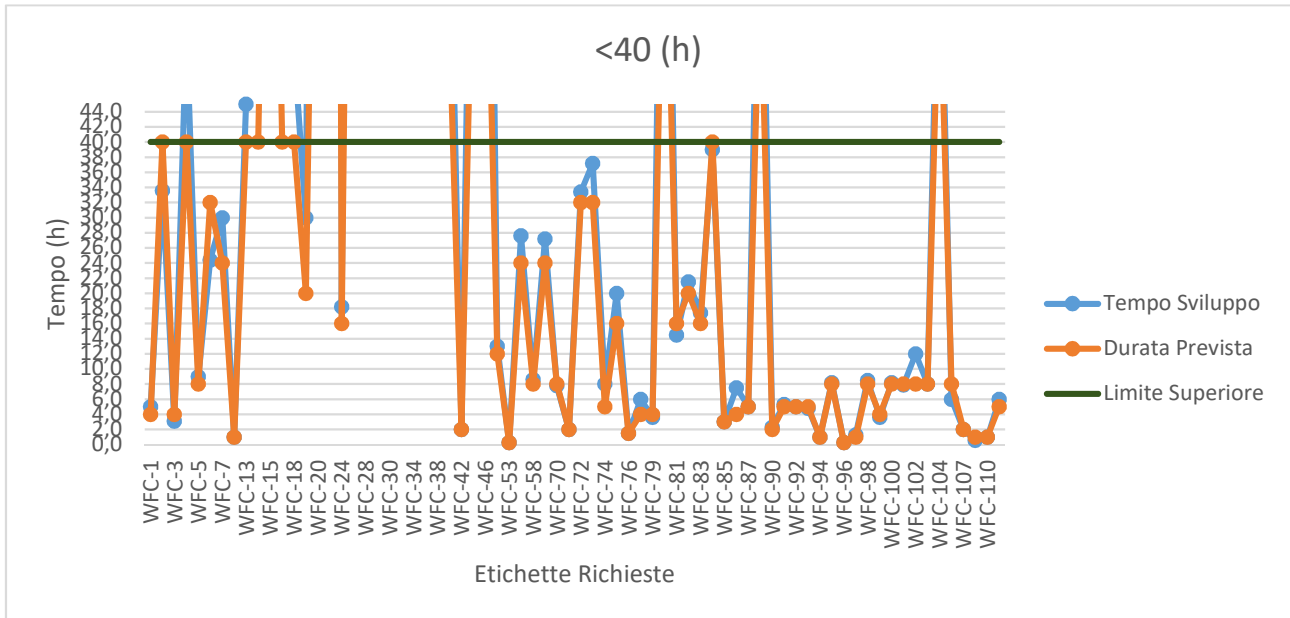
Tempi di sviluppo

La durata di attività delle attività, nella situazione attuale è variabile, essa infatti dipenderà dalla mole di codice da sviluppare e dal numero di attività tecniche da eseguire per ogni richiesta di tipologia CR.

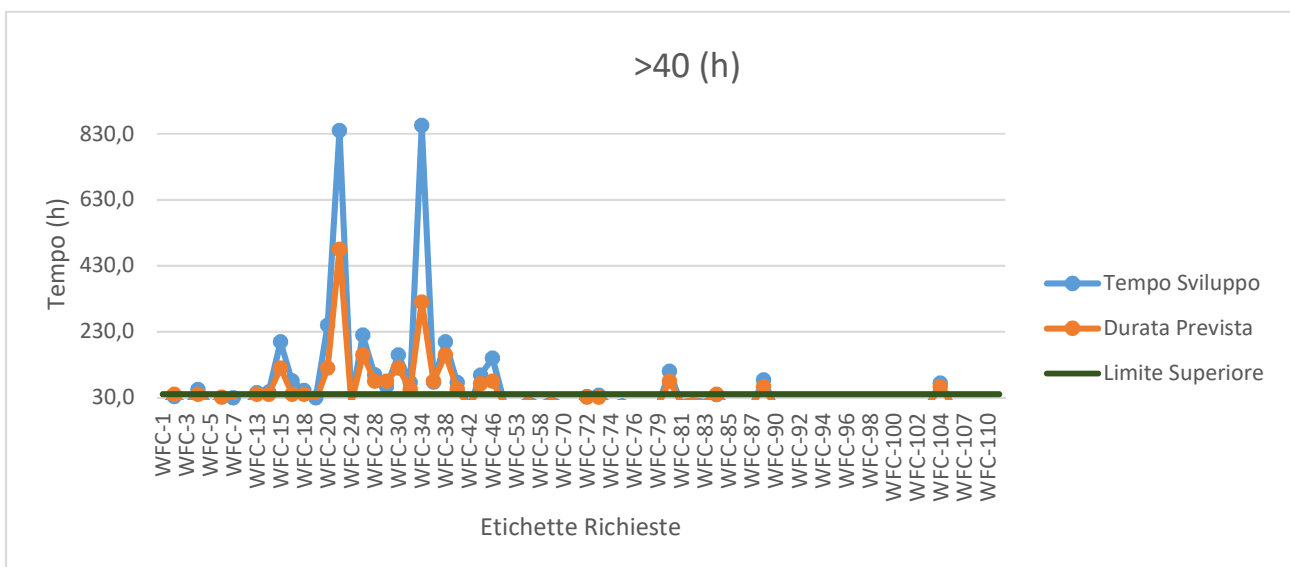
Tuttavia, confrontando il tempo di sviluppo effettivo e la durata prevista di tutte le richieste di sviluppo (CR) e supporto (Service) eseguite nel progetto attuale si è osservato che:



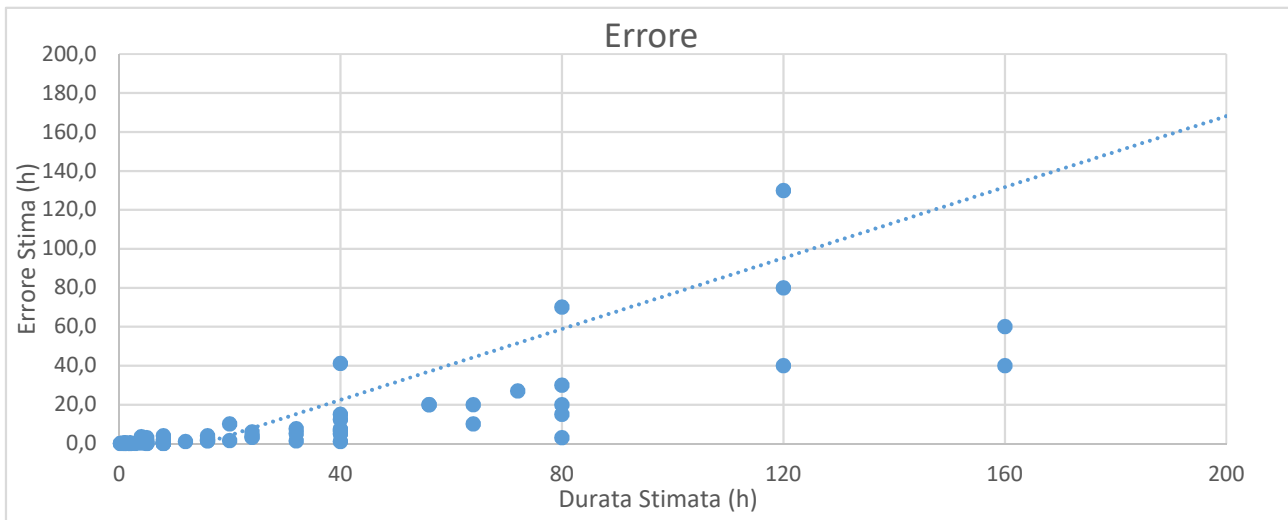
Le richieste, a cui è stata associata una stima preliminare minore di cinque giorni, sono state sviluppate utilizzando un tempo effettivo all'incirca corrispondente alla durata prevista, come è possibile osservare dai grafici sottostanti.



Per quel che riguarda le attività con durata prevista maggiore di cinque giorni, invece, si può osservare che il tempo previsto in fase preliminare si discosta molto dal tempo effettivo di sviluppo delle stesse.



A supporto delle ipotesi precedenti, mediante uno studio più approfondito, si è anche osservato che l'errore di stima, definito come il delta tra il tempo di sviluppo effettivo e la durata prevista, aumenterà in maniera quasi lineare con l'aumentare della durata prevista associata ad una richiesta. In particolare, come osservabile in figura, la precisione della stima delle attività si ridurrà in maniera consistente superata la durata dei cinque giorni.



A seguito delle assunzioni precedenti, nella pianificazione Agile si è ritenuto opportuno suddividere tutte le richieste con durata superiore ai cinque giorni in attività più piccole a cui associare una durata prevista di massimo cinque giorni. In tal modo si è potuto pianificare gli sprint ipotizzando che il tempo di sviluppo effettivo delle stesse sarà corrispondente a quello previsto inizialmente. Infatti possibili errori di stima in difetto non risulteranno significativi e quindi non inficeranno la pianificazione, potendo eventualmente essere compensate da errori in eccesso.

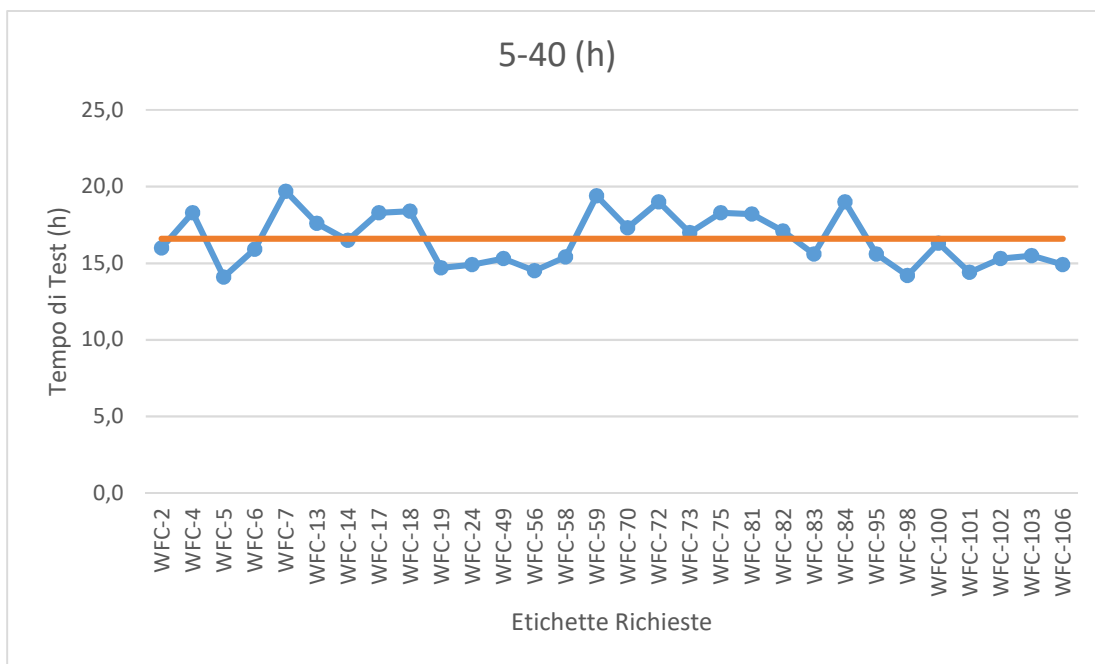
Tempi di Test

La durata dei test può dipendere dall'effort necessario per lo sviluppo della richiesta per via del campione di casi da testare. In caso di numerosità ≥ 2 è possibile che l'ottimo sia non testarli simultaneamente in quanto i programmi su cui risiedono le modifiche necessarie sono sempre gli stessi.

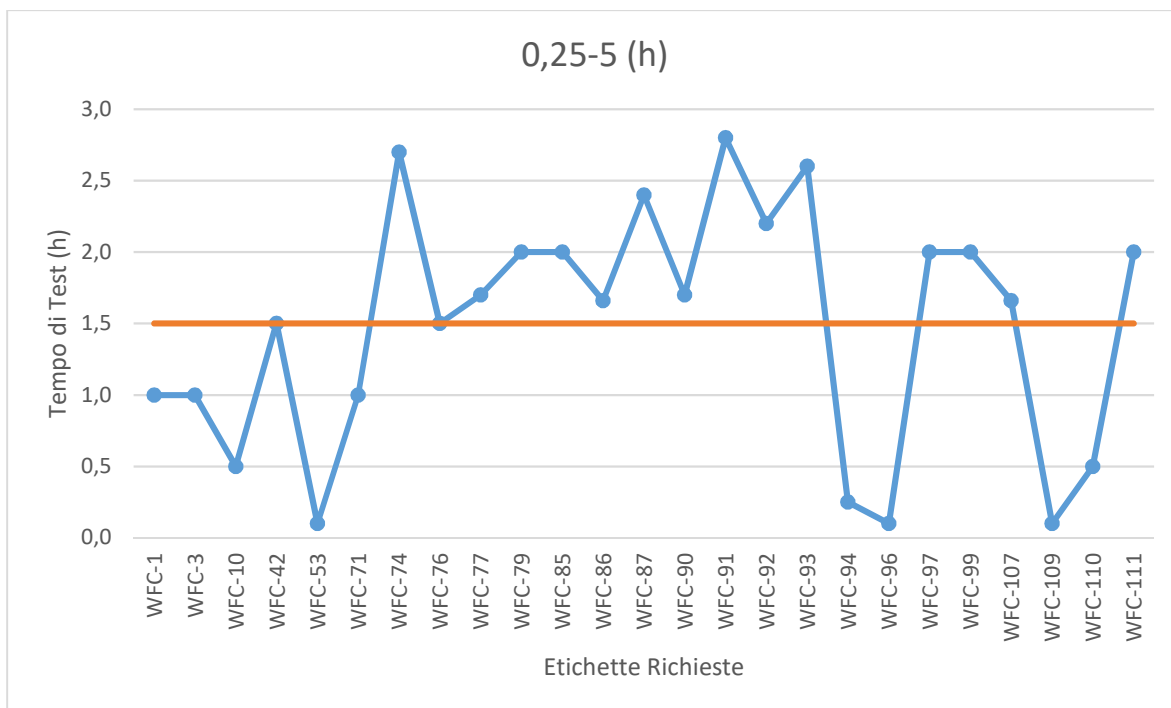
Il test simultaneo non permette di individuare facilmente i bug da sistemare.

Nel dettaglio, dopo un'analisi delle richieste, si è osservato che per le attività con durata prevista di massimo cinque giorni, il tempo medio dei test è incluso nell'intervallo massimo di due giorni. Tali attività di fatto presentano non più di due casi di test da verificare.

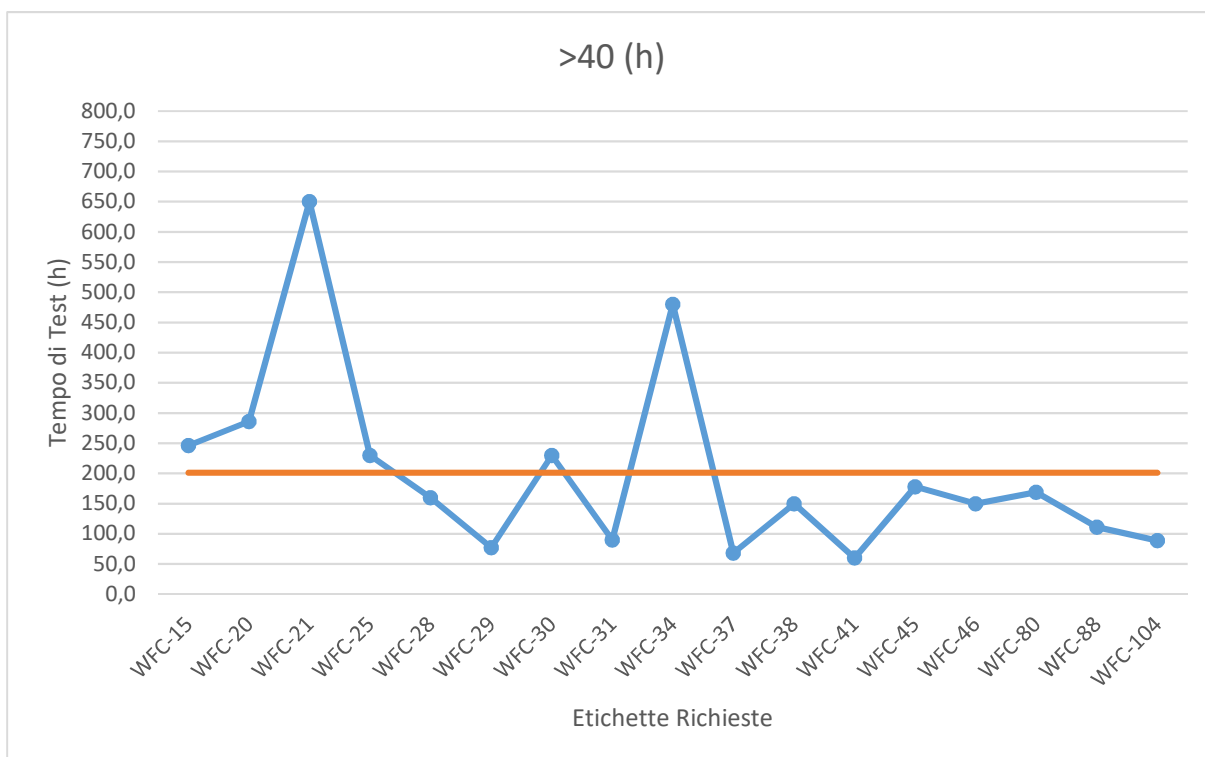
Il tempo di test è in media 2 giorni perché una delle caratteristiche delle attività del recupero crediti è quella di essere interdipendente con l'aggiornamento notturno di AS400, il quale può avviarsi solo a chiusura della giornata lavorativa delle società esterne di recupero conclusasi alle 9 di sera.



Le attività con durata prevista minore di 5 ore si riferiscono ad attività di supporto secondarie che non contemplano il coinvolgimento dell'utente finale, riferendosi ad attività tecniche interne, verificabili unicamente attraverso test interni, ma che non implicano conseguenze sulle restanti attività. Perciò per le attività che ricadono nell'intervallo descritto, il tempo di test non è significativo ai fini della pianificazione degli sprint.



Per quel che riguarda le attività con durata prevista maggiore di cinque giorni, il tempo previsto per i test è variabile, in quanto presentano più di due casi di test con le difficoltà su descritte.



A seguito delle assunzioni precedenti, si è ritenuto opportuno considerare il tempo di test di ogni richiesta pari a due giorni lavorativi.

Il tempo di test, così definito, è stato inserito all'interno della pianificazione degli Sprint in funzione del numero di attività presenti al suo interno, ciò è possibile, in quanto, la durata prevista per tutte le attività definite nella metodologia agile implementata, è al massimo pari a cinque giorni.

4.2.3 Sprint Management

Nella nuova situazione la pianificazione e gestione delle attività avverrà, inizialmente, attraverso la creazione di un Product Backlog e successivamente mediante la programmazione degli sprint di sviluppo, in cui il team di sviluppo, porterà i tasks indicati nello sprint.

Nel dettaglio è necessario specificare come avverrà la gestione di queste due attività, a chi sarà affidato il compito e come è stata implementata la nuova metodologia utilizzando il tool di gestione aziendale (Jira).

Product Backlog

Il Product Backlog rappresenta l'elenco completo delle funzionalità che dovranno essere implementate all'interno dei successivi sprint.

Il Product Backlog è costantemente aggiornato per tutta la durata del progetto/prodotto. Esso include una serie di voci:

- Nuove funzionalità per il cliente: identificate nella nuova gestione sotto forma di User Story, create e gestite dal Product Owner
- Principali obiettivi di miglioramento dell'ingegneria o Lavori di ricerca: gestite attraverso l'apertura di richieste di task direttamente dall'utente, considerate operazioni non creanti inoperatività, ma solo di modifica o aggiornamento.
- Gestione difetti noti: gestite attraverso l'apertura di richieste di task dall'utente in caso di malfunzionamenti, considerate operazioni creanti inoperatività, a cui sarà assegnata la massima priorità e da risolvere al più presto possibile.

Perciò le voci inserite all'interno del Product Backlog possono essere espresse attraverso "User Stories" o "Task" ritenuti adeguati dai principali attori coinvolti nel processo.

Il Product Backlog, affinché sia adeguatamente strutturato, deve soddisfare i seguenti requisiti:

- Dettagliato in modo appropriato; il livello di dettaglio sarà proporzionale alla priorità con la quale le diverse voci verranno elaborate

- Stimato; Il team fornisce al Product Owner le stime dello sforzo per ogni voce del Product Backlog.

La stima può essere effettuata utilizzando diversi approcci: approcci standard prevedono che la stima sia fornita in termini di tempo necessario a svolgere una determinata attività. Sono tuttavia presenti approcci alternativi, fortemente consigliati, che prevedono l'utilizzo di "Story Points".

- Aggiornato; il product Backlog viene continuamente affinato ed aggiornato dal Product Owner tenendo in considerazione aspetti cruciali quali esigenze del cliente, cambiamenti nelle condizioni di mercato, mosse della concorrenza ecc
- Prioritizzato; l'obiettivo perseguito dal Product Owner è quello di fornire il maggior vantaggio economico all'organizzazione ed allo stesso tempo gestire con sufficiente anticipo i rischi elevati

La definizione delle voci, riguardanti lo sviluppo, contenute all'interno del Product Backlog verrà effettuata da parte della nuova figura di Product Owner, definita precedentemente, a seguito di incontri organizzati tra la stessa e gli stakeholders del progetto.

Il Product Owner si occuperà di identificare i requisiti richiesti dall'utente, formalizzandoli attraverso l'apertura di User Story, nella quale sarà contemplata la fase di approvazione del budget.

Le User story aiuteranno il team di sviluppo nell'implementazione e definizione dei vari task tecnici da dover eseguire identificando requisiti, intenzioni, casi d'uso e casi di test.

Per la definizione delle User story, si è analizzato la situazione attuale del progetto soffermandosi sulle richieste definite precedentemente come CR. Tali richieste sono le uniche che necessitano un nuovo sviluppo, quindi verranno aperte e gestite in ottica di soddisfare un requisito del cliente.

Tuttavia, successivamente alla raccolta e definizione dei requisiti, attualmente, non vi è una pianificazione puntuale di tutti i nuovi sviluppi da dover eseguire per soddisfare ogni requisito, ma si parte dallo sviluppo più evidente e necessario, al quale poi ne vengono affiancati iterativamente altri, secondo le varie necessità riscontrate nel tempo al fine di soddisfare lo stesso requisito.

Tutto ciò non permetterà una gestione delle attività organizzata, una stima ben definita e un'idea chiara del numero di attività di sviluppo da eseguire per deliverare un requisito, non potendo in tal modo conoscere preventivamente la data di rilascio del requisito.

In ottica Agile, al fine di poter superare questi problemi, si è pensato di dedicare maggior tempo alla pianificazione preventiva delle attività, partendo da una raccolta dei requisiti, poi ad un'analisi di quest'ultimi da parte del product owner ed infine alla definizione delle attività da dover svolgere per soddisfare tali requisiti.

Il product owner quindi assegnerà una priorità alle User Story e grazie all'aiuto del Team di sviluppo, le suddividerà in attività più piccole e le stimerà nei diversi Sprint planning.

Sprint Planning

Obiettivo: preparazione dello sprint; lo sprint planning è generalmente suddiviso in due parti per comodità.

Parte prima: Product Owner, Team, Scrum Master

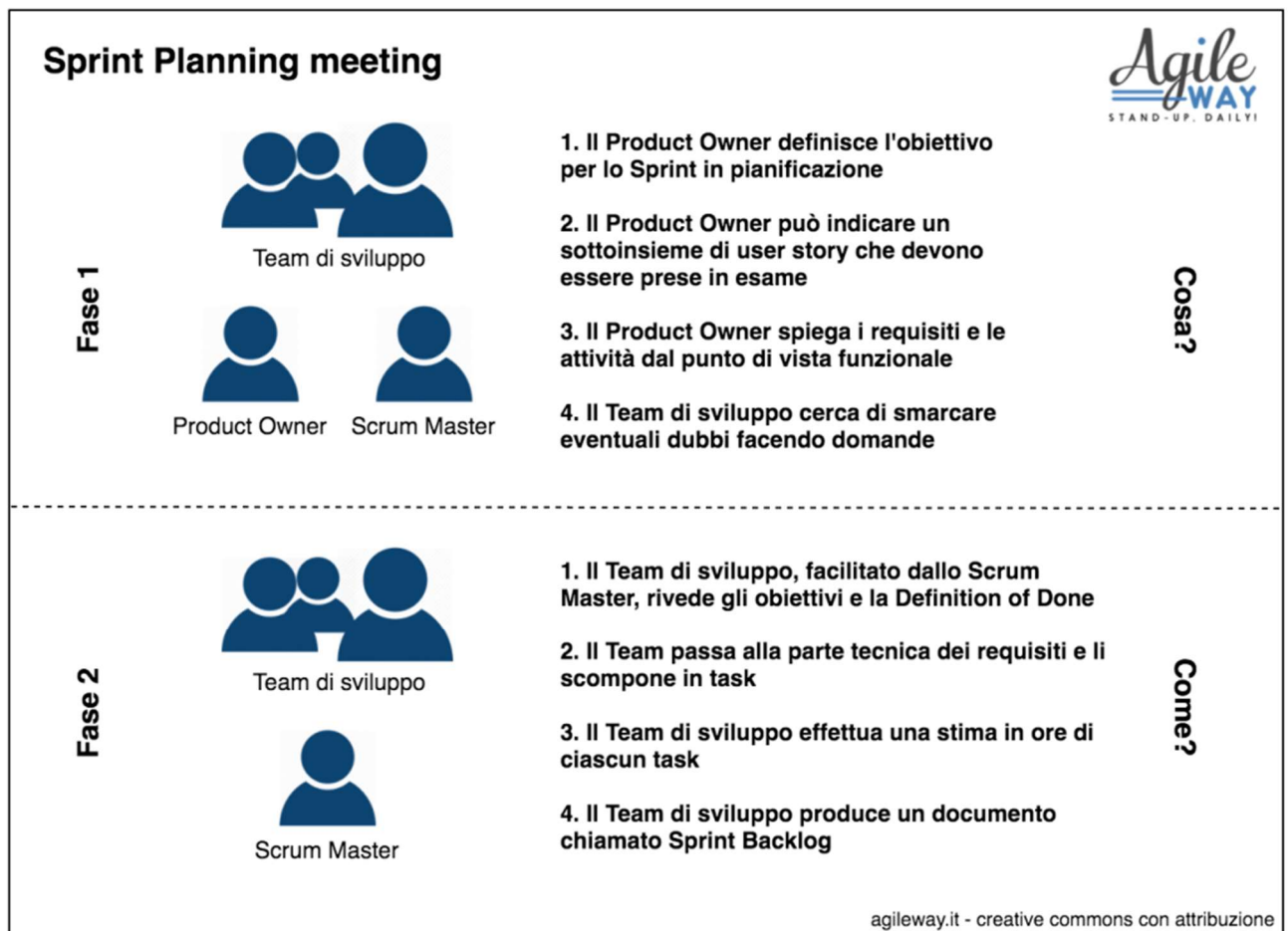
Parte seconda: Team, Scrum Master, Product Owner (facoltativo)

Durata: ogni parte è limitata ad un'ora per ogni settimana prevista per la durata dello Sprint.

Lo sprint Planning sarà effettuato ogni due lunedì, in modo da poter organizzare le attività facenti parte dello sprint successivo di durata di due settimane.

La scelta della durata dello sprint di due settimane è stata fatta in concomitanza dei calendari di rilascio utilizzati attualmente, che prevedono una minor release ogni due settimane e una major a seconda delle necessità e non più di una volta al mese.

In tal modo sarà possibile rilasciare in produzione le nuove caratteristiche in modo dinamico, risolvendo eventuali problemi o effettuando modifiche in modo pressoché immediato, ottenendo un feedback degli utenti continuo.



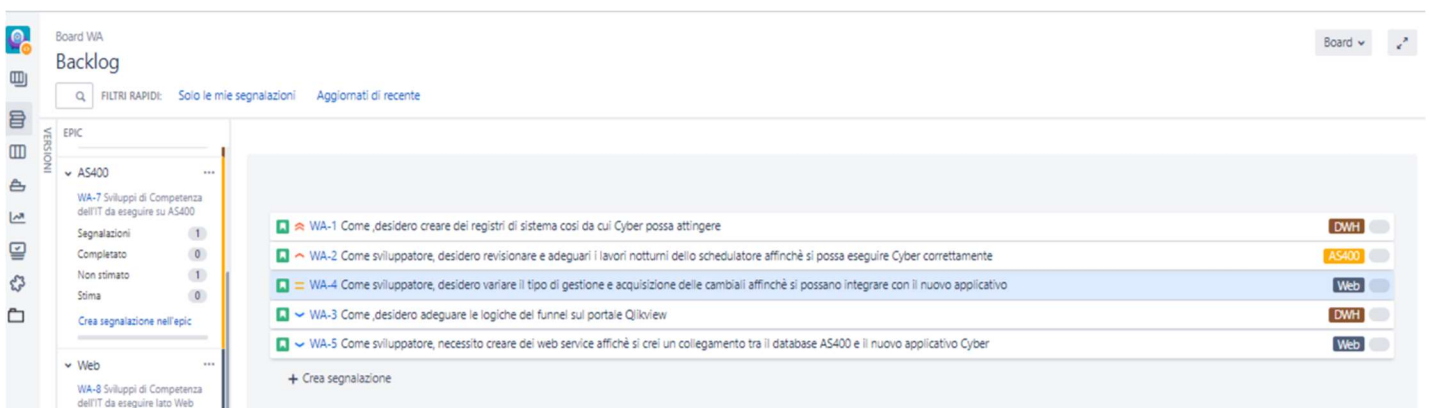
La definizione della quantità di attività che potranno essere svolte all'interno dello sprint successivo può essere effettuata utilizzando approcci differenti:

- Considerare la velocità mostrata all'interno degli sprint precedenti come riferimento su quante attività allocare all'interno dello sprint successivo
- Considerare la capacità del team per lo sprint successivo. Questo approccio richiede che all'interno della seconda parte dello Sprint Planning, ogni membro del team calcoli il tempo dedicabile al prossimo sprint. Le attività verranno inserite all'interno dello Sprint Backlog fino al raggiungimento della capacità individuata dal Team.

Durante lo svolgimento della seconda parte dello Sprint Planning, nella definizione di quelle che saranno le attività svolte all'interno del prossimo sprint, il Team ha il compito di decomporre le voci nel backlog (User Story) in un elenco di singole attività più facilmente gestibili sotto forma di Task.

Qualora le voci del backlog siano già sufficientemente decomposte, esse saranno trasformate in task ed inserite direttamente all'interno dello sprint backlog. Una linea guida comune è che un elemento sia stimato dall'intero Team abbastanza piccolo da potersi completare entro un quarto o meno della durata di uno Sprint.

Come si può osservare in figura, da un'analisi di tutte le richieste di tipologia Cr utilizzate nel progetto attuale, si è ritenuto ideale definire un numero di User Story pari a cinque.



La definizione delle seguenti User story è stata effettuata tenendo conto dei requisiti attuali, facendo una raccolta delle CR aperte inerenti ad un ambito di applicazione o ad implementazioni simili da effettuare. Grazie a questo raggruppamento si è potuta avere anche

una visione più chiara degli attori coinvolti su ogni sviluppo e l'ambito preciso a cui lo stesso fa riferimento. Tale visione non è così chiara nella situazione attuale, a causa della natura incrementale e discontinua delle implementazioni simili, e delle difficoltà di gestione, monitoraggio e controllo del progetto.

In seguito alla definizione delle User Story si è proseguito analizzandole e suddividendole in attività tecniche.

Nel primo sprint planning, ad esempio, si è analizzata la User story con codice "WA-1", alla quale il product Owner ha assegnato la priorità più alta, secondo i vincoli funzionali e temporali definiti.

In collaborazione con il team di sviluppo, l'User story è stata suddivisa in dieci attività.

Successivamente la creazione della User Story, la contestuale suddivisione è stata effettuata prendendo come riferimento l'elenco delle attività simili, utilizzate inizialmente per la sua costituzione, e suddividendole in più attività di minor durata. Ad esse è stata assegnata una stima di massimo 40 ore oltre che un epic di riferimento.

In tal modo tutte le implementazioni e caratteristiche contemplate all'interno delle attuali CR sono state considerate e definite.

The screenshot displays a Jira Backlog board for 'Board WA'. The board is organized into columns for 'AS400', 'Web', and 'Service'. The 'AS400' column contains user stories WA-1 through WA-5. The 'Web' column contains tasks WA-10 through WA-15. The 'Service' column contains tasks WA-16 through WA-19. The right sidebar shows details for the selected task 'WFC Agile / WA-15'.

Task	Estimate	Label
WA-1 Come desidero creare dei registri di sistema così da cui Cyber possa attingere	2.5g	DWH
WA-2 Come sviluppatore, desidero revisionare e adeguare i lavori notturni dello scheduler affinché si possa eseguire Cyber correttamente	2.5g	AS400
WA-4 Come sviluppatore, desidero variare il tipo di gestione e acquisizione delle cambiali affinché si possano integrare con il nuovo applicativo	2.5g	Web
WA-3 Come, desidero adeguare le logiche dei funnel sul portale Qlikview	2.5g	DWH
WA-5 Come sviluppatore, necessito creare del web service affinché si crei un collegamento tra il database AS400 e il nuovo applicativo Cyber	2.5g	Web
WA-10 Creazione registri 301	2g	
WA-11 Creazione registri 302	2g	
WA-12 Creazione Registri 303	2g	
WA-13 Creazione Registri 500	4g	
WA-14 Creazione Registri 600	2.5g	
WA-15 Creazione Registri 601	2.5g	
WA-16 Creazione Registri 700	2.5g	
WA-17 Analisi tecnica Registri 700	2.5g	
WA-18 Creazione Registri 800	2.5g	
WA-19 Analisi tecnica Registri 800	2.5g	

Task Details: WFC Agile / WA-15
Creazione Registri 601
Stima: 2.5g
Rimanente: 2.5g
Stato: APERTA (Mostra Workflow)
Priorità: P3
Componenti: Nessuno
Etichette: Nessuno
Coinvolge la/e Versione/i: Nessuno
Versioni di correzione: Nessuno
Epic Link: Nessuno
Persone: Segnalante: Administrator, Assegnatario: Non assegnata
Dettaglio date: Creato: 04/set/19 11:59, Aggiornato: 04/set/19 12:10

Ad esempio, l'attuale attività WFC-2 di tipo CR, con durata cinque giorni è stata suddivisa in tre attività più piccole (WA-10, WA-11, WA-12) di durata due giorni ciascuna, in modo da rendere la loro gestione, implementazione e verifica più semplice e veloce.

Codice Waterfall	User Storv	Stima originaria	Tipo segnalazione	Riepilogo	Agile	Stima Agile	Sprint Planning	Sprint
WFC-1		4	Service Request	Tracciabilità Registro 200				
WFC-2	WA-1	40	Change Request	Creazione Registri 300	WA-10 WA-11 WA-12	8 8 8	1	1

Allora, definite, stimate e assegnata una priorità alle attività, si è costituito il primo sprint “Sprint1 (Creazione Registri)”.

Per ogni sprint si è definita una capacità massima di 12/15 giorni.

La definizione di tale capacità è stata concepita partendo dall'analisi dei dati attuali, calcolando le ore messe a disposizione da ogni risorsa sul progetto per ogni giorno, come le ore totali di sviluppo effettive impiegate attualmente, diviso i giorni totali impiegati per la conclusione del progetto, a meno dei giorni festivi (Sabato e Domenica); il tutto diviso per il numero di risorse a disposizione:

$$Ore\ Lavorative = \frac{\text{Tempo di sviluppo effettivo}}{(\text{Durata progetto} - \text{Festivi}) \times \text{Numero Risorse}}$$

Il numero di risorse a disposizione è stato definito pari a cinque, in base al numero di risorse mediamente, ad oggi, utilizzato.

Nella definizione dei team di sviluppo, nella metodologia Scrum, sarà necessario prevedere team che siano cross-funzionali ed autonomi. È importante che le attività non siano pre-assegnate durante lo Sprint Planning; l'assegnazione dovrà avvenire in maniera autonoma durante lo svolgimento dello Sprint dalle risorse, a seconda delle necessità.

A questo punto, si potrà definire il primo Sprint, utilizzando un team costituito da 5 membri, considerando due ore di tempo giornaliero a disposizione per ogni risorsa, e trascinando le attività al suo interno, rispettando i vincoli funzionali e di tempo definiti dal product owner e formalizzati tramite l'assegnazione delle priorità alle User Story in primis e poi alle attività

tecniche in cui sono state scomposte, inserendo prima le attività più urgenti con priorità “P1” e poi le meno urgenti con priorità “P5” fino all’esaurimento della capacità massima.

Contemporaneamente alla stima e schedulazione delle attività riferite alle User story, si è gestita anche la schedulazione delle attuali Service, cioè delle attività di supporto alle prime, ma che non necessitano di nuovi sviluppi, ma unicamente di adeguamenti o modifiche di qualcosa di già esistente.

La pianificazione di quest’ultime avvenuta considerando il loro legame con le attività primarie, alle quali sono di supporto, e la loro data di creazione nel progetto attuale, in modo da poter rispettare, anche in questo caso, sia i vincoli funzionali che quelli temporali.

Le attività di supporto che, come tutte le altre attività verranno create, stimate, assegnata loro una priorità all’interno dei vari sprint in fase di Sprint planning nei vari Sprint fino all’esaurimento della capacità massima di quest’ultimi.

The screenshot displays the Jira 'Board WA' interface. At the top, there is a navigation bar with tabs for 'Cruscotti', 'Progetti', 'Segnalazioni', 'Board', 'Time to SLA Plugin', 'BigPicture', 'Power Apps', and 'Crea segnalazione'. Below this, the 'Board WA' section is visible, including a search bar and 'FILTRI RAPIDI: Solo le mie segnalazioni' and 'Aggiornati di recente'. The main area is divided into three sections: 'EPIC', 'Sprint 1 (Creazione Registri)', and 'Backlog'. The 'EPIC' section lists 'AS400' and 'Web' with their respective status counts. The 'Sprint 1' section shows a list of tasks with checkboxes, including 'WA-14 Creazione Registri 600' (2.5g), 'WA-12 Creazione Registri 303' (2g), 'WA-10 Creazione registri 301' (2g), 'WA-13 Creazione Registri 500' (4g), and 'WA-11 Creazione registri 302' (2g). The 'Backlog' section lists tasks like 'WA-1 Come, desidero creare dei registri di sistema così da cui Cyber possa attingere' (DWH), 'WA-2 Come sviluppatore, desidero revisionare e adeguare i lavori notturni dello schedatore affinché si possa eseguire Cyber correttamente' (AS400), 'WA-4 Come sviluppatore, desidero variare il tipo di gestione e acquisizione delle cambiali affinché si possano integrare con il nuovo applicativo' (Web), 'WA-3 Come, desidero adeguare le logiche dei funnel sul portale Qlikview' (DWH), 'WA-5 Come sviluppatore, necessario creare dei web service affinché si crei un collegamento tra il database AS400 e il nuovo applicativo Cyber' (Web), and 'WA-15 Creazione Registri 601' (2.5g). On the right side, there is a 'WFC Agile / WA-14' panel showing details for 'Creazione Registri 600', including 'Stima: 2.5g', 'Rimanente: 2.5g', and 'Stato: APERTA (Mostra Workflow)'. It also lists 'Priorità: P3', 'Componenti: Nessuno', 'Etichette: Nessuno', 'Coinvolge ia/e: Nessuno', 'Versione/i: Nessuno', 'Versioni di correzione: Nessuno', and 'Epic Link: Nessuno'. At the bottom, it shows 'Persone' with 'Segnalante: Administrator' and 'Assegnatario: Non assegnata', and 'Dettaglio date' with 'Creato: 04/set/19 11:59' and 'Aggiornato: 04/set/19 12:29'.

In ottica di una miglior gestione e chiarezza per tutti gli attori coinvolti è stata associata ad ogni attività un epic diverso indicante l’ambito di risoluzione o il tipo di richiesta della stessa:

- DWH: la richiesta utilizza strumenti o effettua modifiche su funzioni del DWH
- AS400: la richiesta utilizza strumenti o effettua modifiche su funzioni di AS400

- WEB: la richiesta utilizza strumenti o effettua modifiche su funzioni proprie dell'ambito Web
- Service: la richiesta è di supporto e non necessita di nuovi sviluppi

In questo modo si è proseguito creando gli sprint successivi con cadenza bisettimanale, fino a conclusione del progetto e quindi al completamento di tutte le attività e requisiti definiti all'interno del Product Backlog.

Si può avere una visione compatta della conversione e suddivisione delle attività attuali (un estratto) in quelle utili all'approccio Agile nella figura sottostante:

Codice Waterfall	User Story	Stima originaria	Riepilogo	Codice Agile	Stima Agile	Tempo Test	Sprint Planning	Sprint
WFC-1	Supporto	4	Tracciabilità Registro 200	WA-20	4	0	1	1
				WA-10	8	16		
WFC-2	WA-1	40	Creazione Registri 300	WA-11	8	16	1	1
				WA-12	8	16		
WFC-3	Supporto	4	Predisposizione tracciato 200	WA-21	4	0	1	1
			Adeguamento Registro Bianco	WA-22	20	16		1
WFC-4	Supporto	40	Tracciabilità e Analisi Tecnica Registro	WA-23	20	16	1	3
			300	WA-24	8	16	1	2
WFC-5	Supporto	8	Creazione Registri 500	WA-13	32	16	1	2
WFC-6	WA-1	32	Estrazione dei job notturni CBU e prima	WA-25	16	16	2	3
			analisi IT degli stessi	WA-26	8	0		3
WFC-7	Supporto	24	Creazione utenza Jira per Letizia	WA-27	1	0	3	3
			Alviano					
WFC-10	Supporto	1	Analisi tecnica e tracciabilità Registri	WA-28	20	16	1	3
			600	WA-29	20	16		4
WFC-13	Supporto	40	Creazione Registri 600	WA-14	20	16	1	1
				WA-15	20	16		2
				WA-68	24	16		15
WFC-14	WA-1	40	Creazione Maschere di valutazione	WA-69	24	16		15
			accettazione Cambiali e Piani di	WA-70	24	16	7	16
WFC-15	WA-4	120	Rientro	WA-71	24	16		16
				WA-72	24	16		16
WFC-17	WA-1	40	Creazione Registri 700	WA-17	20	16	2	3
				WA-16	20	16		2
WFC-18	WA-1	40	Creazione Registri 800	WA-18	20	16	2	2
				WA-19	20	16		3
WFC-19	Supporto	20	Modifica tracciato 200	WA-67	20	16	4	4
				WA-30	24	16		4

Nel dettaglio, all'interno della figura, è possibile osservare lo sprint planning e lo sprint in cui ciascuna attività è stata analizzata e pianificata, la relazione tra le User Story e le vecchie CR e le attività di durata minore in cui le User story, con riferimento alle vecchie CR, sono state suddivise nel nuovo approccio.

● Daily Scrum

Obiettivo: Aggiornamento e coordinamento tra i membri del Team

Partecipanti: Team, Product Owner (se richiesto), Scrum Master

Durata: 15 minuti al massimo

In ottica di monitoraggio e miglioramento continuo, sarà di fondamentale importanza il Daily scrum.

Il Daily Scrum rappresenta un breve incontro (massimo 15 minuti), svolto quotidianamente ad un orario prestabilito.

Il meeting rappresenta un'occasione per tutti i membri del Team per aggiornarsi, sincronizzare il proprio lavoro ed informare i colleghi riguardo la presenza di eventuali ostacoli. Durante lo svolgimento del meeting, ogni membro del Team dovrebbe aggiornare i colleghi riguardo:

- Cosa è stato fatto dopo l'ultima riunione
- Che cosa sarà fatto entro la prossima riunione
- Quali ostacoli ci sono da affrontare

Al fine di rendere autonoma e adattiva alle necessità la struttura organizzativa, si è preferito non stabilire un orario predefinito per il Daily Scrum. Infatti, trattandosi di un meeting di durata ridotta, lo scrum master potrà flessibilmente, giorno per giorno, definirlo a seconda di esigenze, imprevisti e comodità.

Sprint Review

Obiettivo: Ispezione e adeguamento relativi all'incremento di funzionalità di prodotto

Partecipanti: Team, Product Owner, Scrum Master, ed in generale tutti i soggetti interessati

Durata: limitata ad un'ora per ogni settimana di durata dello Sprint

La Sprint Review è un'attività di ispezione ed adeguamento del prodotto. Durante lo svolgimento del meeting è consentito l'utilizzo del software e delle nuove funzionalità implementate all'interno dello Sprint.

È importante che vi sia una conversazione tra il Team ed il Product Owner con l'obiettivo di incrementare la comprensione della situazione, fornire input e suggerimenti utili per le successive migliorie.

Si è ipotizzato di definire lo Sprint Review il lunedì immediatamente successivo all'avvio dello sprint, in ottica di monitoraggio così da poter verificare in corso le attività già svolte nello sprint e poter eventualmente definire approcci di risoluzione differenti per le attività ancora da fare.

Perciò lo sprint review si terrà, come l'avvio dello sprint, a cadenza bisettimanale.

Retrospettiva di Sprint

Obiettivo: Ispezione ed adeguamento riguardanti il processo e l'ambiente

Partecipanti: Team, Scrum Master, Product Owner

Durata: Limitata a 45 minuti per settimana di durata dello Sprint

La Retrospettiva di Sprint, che segue la Sprint Review, è un'attività di ispezione ed adeguamento del processo e dell'ambiente.

L'evento in esame rappresenta l'occasione per discutere riguardo alle modifiche che potrebbero essere messe in atto al fine di migliorare il processo.

È importante, che, oltre agli aspetti negativi, vengano messi in luce i punti di forza e gli aspetti positivi individuati durante lo svolgimento delle attività.

A conclusione della Retrospettiva di Sprint, la metodologia Scrum prevede che venga iniziato un nuovo Sprint, ripetendo in maniera ciclica gli eventi descritti all'interno del presente capitolo.

Infatti si è ritenuto utile definire l'ultimo venerdì di ogni sprint come giorno utile per fissare la retrospettiva dello sprint, in modo da discutere immediatamente le difficoltà riscontrate e poterle risolvere già all'avvio del prossimo sprint in partenza il lunedì successivo.

4.2.4. Un Nuovo Modo di gestire la conoscenza: “Confluence”

Nella situazione AS-IS, uno dei principali problemi riscontrati è legato alla documentazione. Come descritto precedentemente, ad oggi, i membri appartenenti al Team dei Demand hanno il compito di redigere quantità elevate e spesso ripetitive di documenti funzionali (DDR) legati ad ogni richiesta di sviluppo(CR), allegati a quest’ultima e non immagazzinati all’interno di uno spazio comune, al massimo conservati in diverse cartelle condivise in rete, non facilmente raggiungibili e nel quale non è immediato ricercarli.

La struttura organizzativa attuale di conservazione non permette di definire facilmente una sequenza cronologica dei diversi documenti e la presenza di più versioni degli stessi, causandone duplicati.

La documentazione che risulta non uniforme e standardizzata per tipo o ambito di appartenenza, a causa della mancanza di modelli unici da precompilare, di processi chiari e definiti di formalizzazione della conoscenza implicita e di un supporto tecnologico adeguato.

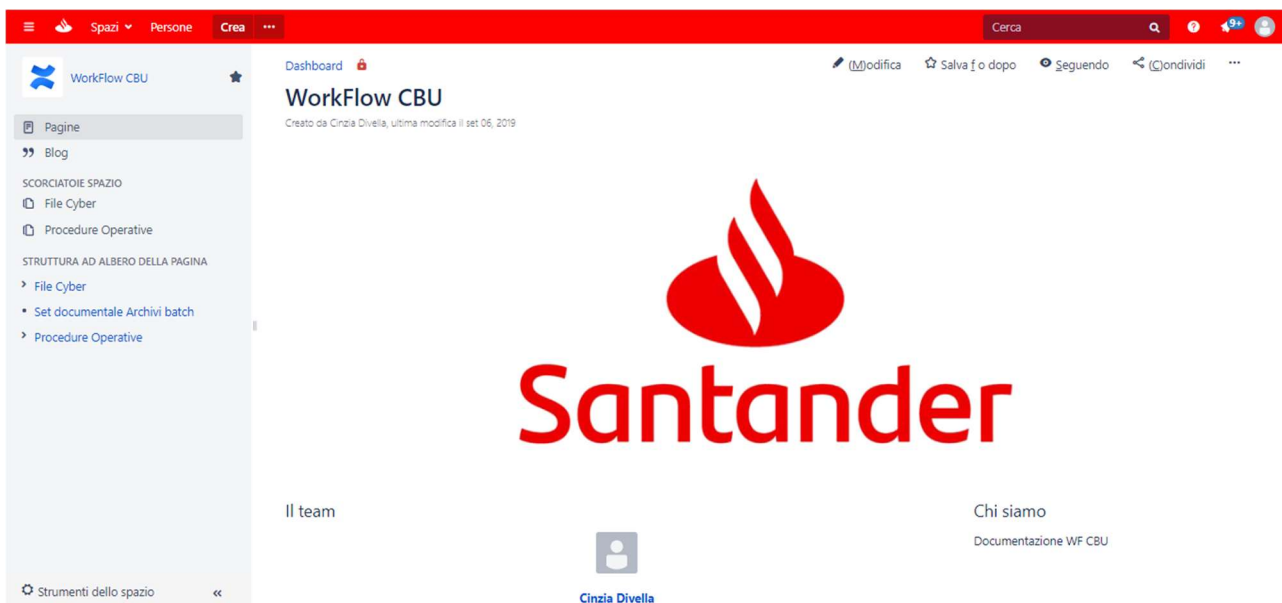
La gestione della conoscenza, dunque, è una questione delicata per la situazione attuale.

In concomitanza con l’introduzione di una nuova metodologia di gestione del progetto, si è pensato quindi di introdurre un nuovo applicativo tecnologico, mediante il quale poter gestire anche la conoscenza in maniera Agile.

In ottica di integrazione e continuità si è scelto di acquistare l’applicativo di Atlassian (Confluence) grazie al quale poter gestire trasversalmente la documentazione necessaria ad un progetto e alla risoluzione di eventuali problemi, invece tracciati, pianificati e gestiti sull’applicativo di trouble ticketing di Atlassian (Jira) già presente in azienda.

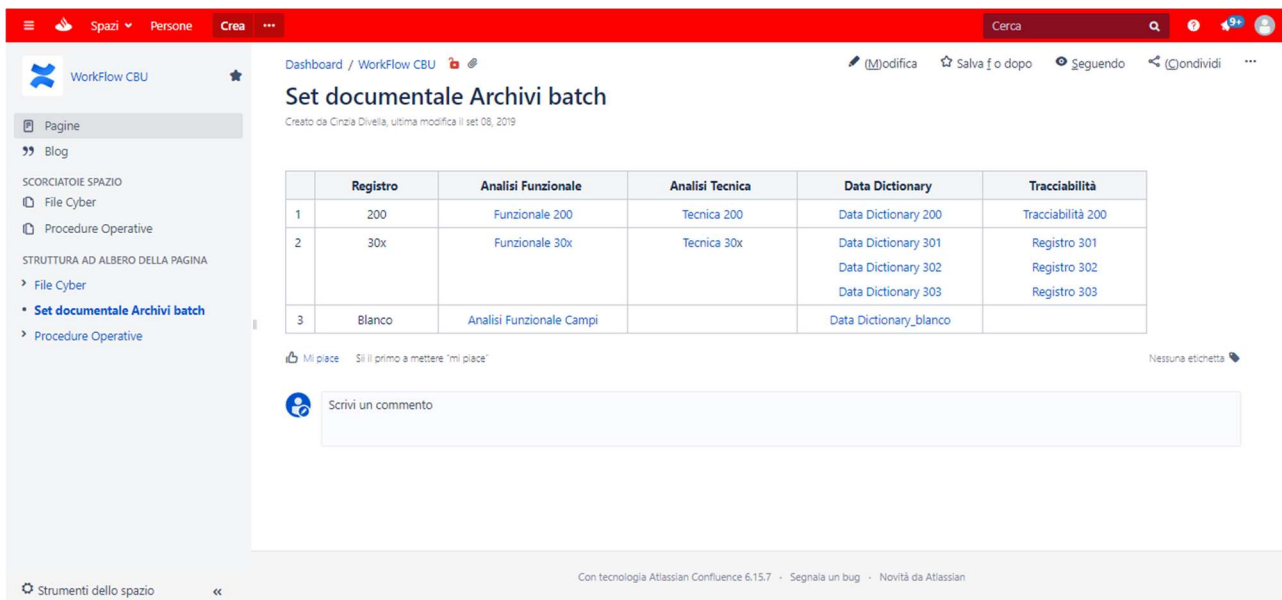
Perciò si è ipotizzato di gestire la documentazione e quindi la conoscenza derivante dal progetto “Workflow CBU”, oggetto di analisi della tesi, utilizzando l’applicativo (Confluence) introdotto.

Si è creato, quindi, all’interno di Confluence, uno spazio comune nominato come il progetto, nel quale poter inserire analisi tecniche, analisi funzionali, allegati e procedure operative e di risoluzioni di eventuali problemi.



Nel dettaglio sono stati censiti all'interno della sezione "Set documentale archivi Batch", tutti i registri necessari all'implementazione e funzionamento dell'applicativo Cyber Financial.

In particolare ad ogni registro è stata allegata, facilmente, la corrispettiva analisi funzionale, redatta da i Demand del progetto, e l'analisi tecnica, di competenza del team di sviluppo.



La creazione dei registri, inoltre, ha necessitato, per uniformità di linguaggio e standard di utilizzo, la creazione di Data Dictionary, in cui specificare, per ogni registro, il nome originale del campo da voler inserire nel flusso risultante dall'elaborazione dell'applicativo ed il nome

di riferimento utilizzato all'interno dello stesso. In tal modo si potrà avere un chiaro riferimento e dettaglio di tutti gli elementi, senza la presenza di ambiguità, incomprensioni ed errori.

In ottica di tracciabilità, invece, è risultato utile formalizzare i processi di analisi, acquisizione e elaborazione dei dati gestiti da ogni registro all'interno del corrispettivo documento indicato nella sezione intitolata proprio "Tracciabilità".

In questo modo, quindi, si potrà denotare la facilità di ricerca ed organizzazione di tutti gli elementi necessari, in primis alla creazione dei registri, ma poi al progetto, conservati all'interno di uno spazio comune, organizzati in maniera puntuale per numero di registri e intuitivamente analizzabili con un semplice click.

Documentazione che potrà essere standardizzata ed unificata mediante l'utilizzo di modelli predefiniti o resi tali dall'azienda utili agli utenti nella scrittura delle procedure e analisi.

Il tutto reso ancora più intuitivo, dinamico e veloce grazie alle ulteriori funzioni di ricerca e modifica in tempo reale offerte dallo strumento.

Infatti sarà possibile ricercare mediante parole chiave o etichette i file desiderati e modificarli in tempo reale ed in maniera cooperativa da tutti gli utenti abilitati.

La visualizzazione e modifica delle pagine che potrà, ancora, essere permessa o meno grazie alle restrizioni fornite dallo stesso applicativo.

L'applicativo potrà anche essere utilizzato a supporto del sistema di Trouble ticketing, utilizzato per la gestione delle richieste del progetto, in ottica monitoraggio delle stesse, potendone monitorare andamento, puntualità di esecuzione e stato.

È stata dedicata una pagina per ogni sprint nel quale è indicato il numero e l'etichetta delle richieste contenute nello stesso, il loro stato corrente, automaticamente aggiornato grazie all'integrazione e al collegamento con Jira, e tutti i grafici di review e monitoraggio offerti direttamente dal tool per la gestione Agile, ad esempio il grafico Burndown.

Un esempio nella figura sottostante:

Confluence Test Spazi Persone Crea

Dashboard / Workflow CBU / WorkflowAgile 10 collegamenti JIRA

Cerca

Modifica Salva | o dopo Seguendo Condizionali

Sprint 1 (05/02/2018-16/02/20)

Creato da Administrator, ultima modifica un momento fa

Data	Feb 5, 2019-Feb 16 2019
Issue	7 issues
Stima Durata	12 giorni

Riepilogo
Il seguente Sprint ha lo scopo di predisporre e creare nel sistema i registri necessari all'elaborazione dei dati da parte dell'applicativo "CyberFinancial"

Punti salienti di questo rilascio

1. Creazione registri
2. Tracciabilità
3. Adeguamento Registri

Tutti gli aggiornamenti per questa releaseEpic

- WA-24 **DA COMPLETARE** Service
- WA-23 **DA COMPLETARE** DWH

Attività

Elenco Richieste	Epic	UserStory
<input checked="" type="checkbox"/> WA-10 - Creazione Registri 301 DA COMPLETARE	<input checked="" type="checkbox"/> WA-23 - DWH DA COMPLETARE	<input checked="" type="checkbox"/> WA-1 - Come 'desidero creare dei registri di sistema così da cui Cyber possa attingere DA COMPLETARE
<input checked="" type="checkbox"/> WA-11 - Creazione Registri 302 IN CORSO	<input checked="" type="checkbox"/> WA-23 - DWH DA COMPLETARE	<input checked="" type="checkbox"/> WA-1 - Come 'desidero creare dei registri di sistema così da cui Cyber possa attingere DA COMPLETARE
<input checked="" type="checkbox"/> WA-12 - Creazione Registri 303 DA COMPLETARE	<input checked="" type="checkbox"/> WA-23 - DWH DA COMPLETARE	<input checked="" type="checkbox"/> WA-1 - Come 'desidero creare dei registri di sistema così da cui Cyber possa attingere DA COMPLETARE
<input checked="" type="checkbox"/> WA-14 - Creazione Registri 600 IN CORSO	<input checked="" type="checkbox"/> WA-23 - DWH DA COMPLETARE	<input checked="" type="checkbox"/> WA-1 - Come 'desidero creare dei registri di sistema così da cui Cyber possa attingere DA COMPLETARE
<input checked="" type="checkbox"/> WA-24 - Predisposizione tracciato 200 FATTO	<input checked="" type="checkbox"/> WA-24 - Service DA COMPLETARE	Supporto
<input checked="" type="checkbox"/> WA-22 - Adeguamento Registro Bianco (aggiungere le informazioni sui tassi) DA COMPLETARE	<input checked="" type="checkbox"/> WA-24 - Service DA COMPLETARE	Supporto

Diagramma di burndown

Confronto tra il tempo rimanente per completare le attività e il tempo rimanente per il sprint. La linea rossa rappresenta il tempo rimanente per completare le attività, e la linea verde rappresenta il tempo rimanente per il sprint. La linea grigia rappresenta il tempo rimanente per il sprint.

Diagramma di burndown

Confronto tra il tempo rimanente per completare le attività e il tempo rimanente per il sprint. La linea rossa rappresenta il tempo rimanente per completare le attività, e la linea verde rappresenta il tempo rimanente per il sprint. La linea grigia rappresenta il tempo rimanente per il sprint.

Scrivi un commento

LICENZA SVILUPPATORE - Questo sito Confluence è solo per scopi non produttivi.
Con tecnologia Atlassian Confluence 6.15.6 - Segnala un bug - Novità da Atlassian

ATLASSIAN

Capitolo 5: Analisi dei KPI

Nel seguente paragrafo si analizzeranno i risultati ottenuti a seguito sia della pianificazione del progetto “Workflow CBU” secondo la nuova metodologia agile, sia dell’applicativo tecnologico (Confluence) introdotto per la gestione della conoscenza.

Nel dettaglio potremo osservare che, con l’introduzione di una nuova metodologia, è stato necessario affrontare le difficoltà dovute al maggior sforzo nella pianificazione sia da parte del product owner che del team di sviluppo, non più limitato unicamente ad eseguire gli sviluppi assegnati ma attivamente coinvolto nelle attività di stima, pianificazione e review del progetto.

Il Team di sviluppo che, come descritto precedentemente, dovrà rivoluzionare il proprio modo e mentalità di lavorare, diventare autonomo, partecipe e responsabile della tempestiva esecuzione delle attività all’interno dello sprint, auto assegnandosi le attività e gestendole nel tempo prefissato dallo sprint secondo la capacità massima di tempo messa a disposizione per il progetto.

Ulteriori difficoltà che potranno essere riscontrate nella fase di transizione tra la metodologia attuale e quella Agile, si sostanziano nel cambio di ruolo e definizioni della tipologia di richiesta da dover utilizzare.

Tali difficoltà tuttavia si riferiscono maggiormente ad un cambio di mentalità e quindi alla fase di transizione iniziale, che, se gestita correttamente dallo Scrum master, potrà essere superata senza troppi problemi ed in poco tempo.

La nuova gestione agile del progetto ipotizzata e l’introduzione del Knowledge management, ad eccezione degli ostacoli evidenziati, dimostrerà invece benefici di diversa natura: tempo, costo e qualità.

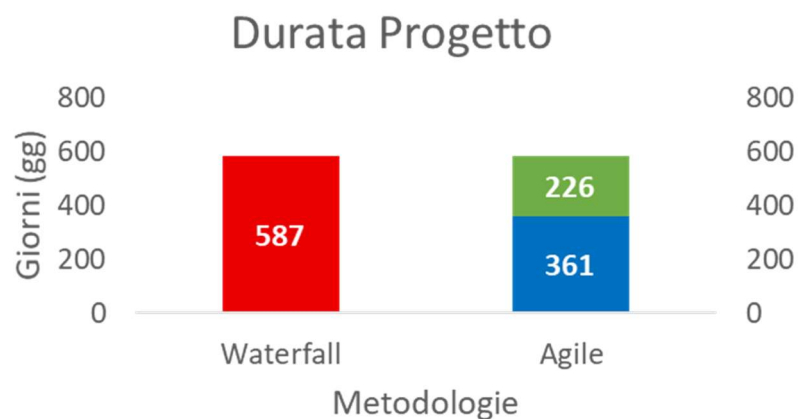
La verifica dei benefici proposti nei prossimi paragrafi si potrà dimostrare in seguito alla messa in pratica della gestione agile nei progetti futuri, di natura, dimensioni e caratteristiche simili del progetto analizzato, anche scalandoli e valutandoli sull’intero portfolio di progetti.

5.1 Tempo

In seguito alla pianificazione e scomposizione delle attività ipotizzata e mediante un'analisi più dettagliata dei tempi, si sono potuti evincere i seguenti risultati:

1. Durata Progetto:

La scomposizione delle attività, la loro stima, pianificazione negli sprint, esecuzione e test delle stesse, ha necessitato l'esecuzione di venticinque Sprint per la realizzazione del progetto. In particolare avendo impostato la durata di ogni sprint a due settimane, si è potuto osservare che rispetto alla durata del progetto gestito nelle modalità attuali, si sono riscontrati miglioramenti notevoli di tempo con la gestione Agile.



	Waterfall		Agile	
	Inizio	Fine	Inizio	Fine
Durata Progetto	05/02/2018	15/09/2019	05/02/2018	01/02/2019

Grazie allo sviluppo Agile, il progetto attualmente di durata 587 giorni, si potrà completare in soli 361 giorni circa, ottenendo così una riduzione pari a 226 giorni, tramutabili in un risparmio effettivo, eliminando i giorni festivi (Sabato e Domenica), di 1360 h.

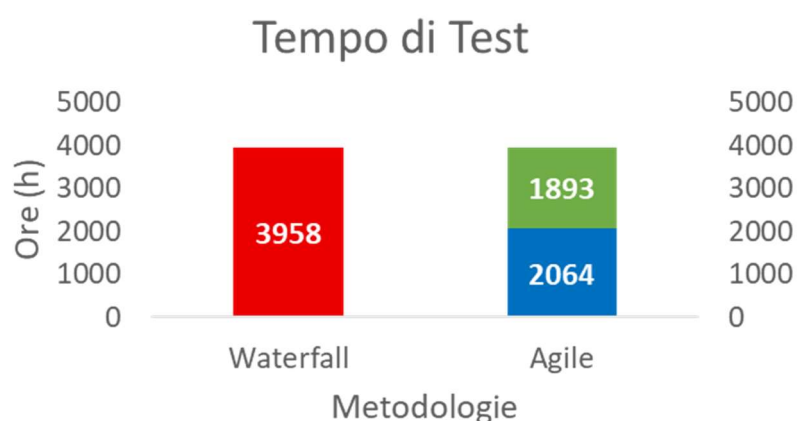
Tale riduzione, come vedremo di seguito, sarà dovuta principalmente alla riduzione di tempo e del numero delle singole fasi delle richieste da gestire per il completamento del progetto e porterà anche risparmi in termini di costi.

2. Tempo di Test

A supporto dei risultati evidenziati prima per la riduzione della durata del progetto, sono stati analizzati anche i tempi delle singole fasi che costituiscono il progetto e quindi la sua durata.

La prima osservazione si è focalizzata sul tempo dedicato per ogni richiesta nell'effettuare la verifica degli use cases che la richiesta deve sviluppare e portare a termine.

Anche in questo caso si è rilevata una riduzione cospicua dei tempi richiesti per i test.



	Waterfall	Agile
Tempo per test (h)	3958	2064

Nel dettaglio, la metodologia Waterfall attuale ha richiesto 3958 ore circa di test, che grazie allo Scrum si sono ridotte a 2064 ore, ottenendo un risparmio di circa 1893 ore pari al 47% del tempo iniziale.

In particolare, questa riduzione è stata possibile grazie alla suddivisione di ogni attività in più attività che prevedevano al massimo una durata prevista di cinque giorni e la verifica di un solo test case, che come dimostrato nelle assunzioni fatte precedentemente, richiedono in media un tempo definibile di due giorni.

Infatti nella gestione precedente, le attività, avendo una durata prevista maggiore di 5 giorni e più test case da verificare, non permetteva la definizione del tempo richiesto per i test, anche a causa delle molteplici caratteristiche da testare e la confusione creatasi, in caso di problemi, nell'associazione di quest'ultime ad una causa sorgente.

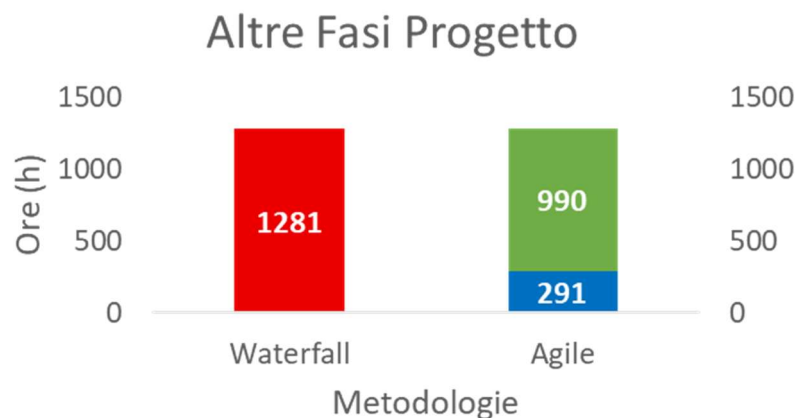
3. Fasi di progetto

La seconda analisi si è concentrata invece sul tempo impiegato per gestire le altre fasi e stati che ogni richiesta deve assumere nell'arco del suo ciclo di vita, ad esclusione di quelli citati precedentemente.

Da un'osservazione qualitativa preliminare si può immediatamente notare come il numero di stati che compongono i Workflows delle richieste, grazie alla gestione Agile, si sia notevolmente ridotto, passando da flussi che comprendevano 29 stati a flussi semplificati di 6.

Da ciò si può subito comprendere come la gestione delle richieste sarà più rapida, dinamica e flessibile, in sostanza facilitata, da cui ne deriveranno riduzioni di impiego di tempi, costi e risorse.

In ottica di tempi si è potuto dimostrare come il tempo dedicato alla gestione di tutte queste fasi si sia ridotto da circa 1281 ore a sole 291 ore, le quali, nel dettaglio sono state calcolate prendendo in considerazione le ore dedicate agli eventi (Sprint Planning, Sprint Review, Daily scrum e Retrospective) precedentemente descritti, unici momenti di gestione al di fuori dei tempi di sviluppo e test, per una riduzione totale di circa 990 h pari al 77% del tempo dedicato inizialmente.



	Waterfall	Agile
Tempo per altre fasi (h)	1281	291

In tal modo dunque l'impiego delle risorse aventi la responsabilità su queste altre fasi si ridurrà notevolmente, riducendone costi e creando nuove opportunità di guadagno mediante l'acquisizione e gestione di nuovi progetti.

4. Pianificazione e Stima

La terza analisi ha preso in esame i tempi e le modalità di pianificazione e stima delle attività nei due metodi di gestione a confronto.

Nella situazione attuale i tempi di pianificazione e stima delle attività non sono definiti puntualmente nel Gantt delle risorse, essendo sostanzialmente effettuati in riunioni occasionali ed il cui tempo è variabile in base alla disposizione dei partecipanti, causandone slittamenti di attività non previsti in principio o ritardi di consegna.

Grazie all'agile si è potuto definire e considerare i tempi di stima e pianificazione nei vari sprint planning tenutisi. Infatti all'interno di quest'ultimi il product owner e il team di sviluppo dovranno in collaborazione decidere quali attività è necessario svolgere, in quanto tempo e in che sprint eseguirle.

Per quanto riguarda la stima, si è anche potuta evincere una maggior precisione della previsione della stessa rispetto al tempo di sviluppo, dovuta, come dimostrato nelle assunzioni del capitolo precedente ad una divisione più dettagliata delle attività richiedenti una durata prevista minore.

Precisione della stima che nella modalità di gestione attuale è risultata uno dei fattori più critici, pari all'incirca allo 0,83.

La precisione è stata calcolata nel seguente modo:

$$0 < Precisione = \frac{Durata\ Prevista}{Tempo\ di\ sviluppo\ effettivo} < 1$$

Cioè tramite il rapporto della durata prevista, cioè stimata inizialmente, ed il tempo di sviluppo utilizzato per l'esecuzione dell'attività.

L'indicatore risulterà tanto più ottimale quanto più tenderà ad 1, viceversa per lo zero. Infatti quest'ultimo avrà un dominio compreso tra zero ed uno.

In particolare, nel caso in cui la durata prevista risulti maggiore del tempo di sviluppo, l'indicatore risultante sarà maggiore di 1. Per ottenere un valore comparabile, si dovrà considerare la funzione reciproca (1/precisione). Ci si aspetta che l'indicatore sia molto più vicino all'1 grazie alla modalità Agile.

5.2 Costi

Nel seguente paragrafo saranno analizzati i benefici derivanti dalla gestione Agile del progetto in ottica di costi.

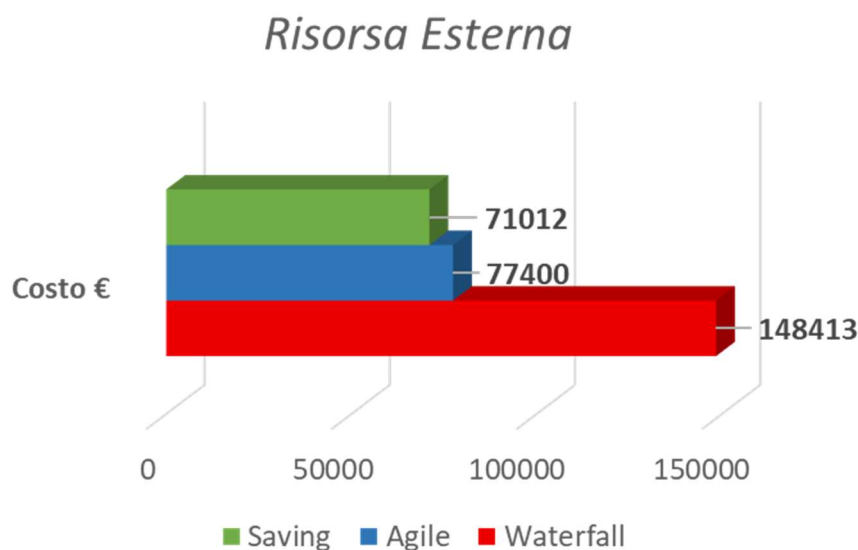
Nel progetto considerato si sono anche riscontrati delle riduzioni di costi oltre ad un beneficio in termini di tempi e frequenza di rilascio di valore.

Tutto ciò, possibile anche grazie all'ipotesi iniziale sul tempo di sviluppo e numero di risorse utilizzate pari alla situazione attuale Waterfall.

Infatti i costi legati alla fase di sviluppo non varieranno, ma i benefici di costo riscontrati saranno dovuti ai risparmi di tempo ottenuti dal non impiego delle risorse, interne ed esterne, legate alla gestione delle altre fasi, come vedremo meglio nel prossimo paragrafo.

1. Risorse esterne

Nel dettaglio si potrà associare al risparmio di tempo utilizzato per eseguire le verifiche degli use case un risparmio economico pari a 71012 €, circa 47% dei costi iniziali.



	Waterfall	Agile
Costo medio risorsa esterna (€)	37,5	37,5
Costo totale (€)	148413	77400

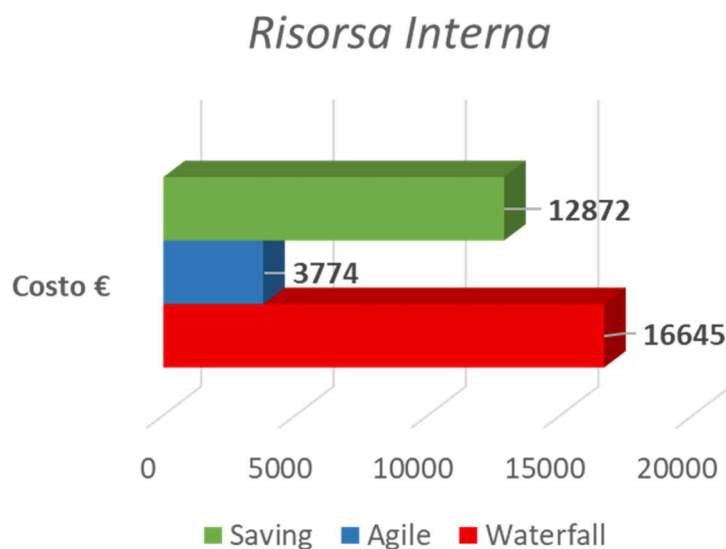
Calcolalo come il prodotto del costo medio orario di una risorsa esterna per le ore di test risparmiate e precedentemente calcolate.

Si è preso in considerazione il costo medio di una risorsa esterna, in quanto l'azienda utilizza di norma delle società di consulenza esterne per l'esecuzione dei test e la verifica dei test case. Società con cui stipula contratti annuali per risorsa di circa 80000 € lordi.

Il costo orario è stato calcolato, dunque, a partire dal costo annuale, considerando 12 mesi composti da 22 giorni l'uno ed 8 ore giornaliere lavorative.

2. Risorse interne

In riferimento al risparmio di tempo legato alla gestione delle altre fasi, in particolare quelle di stima e pianificazione, i costi potranno essere ridotti di circa 12872 euro, pari al 77% dei costi iniziali.



	Waterfall	Agile
Costo medio risorsa interna (€)	13	13
Costo totale (€)	16645	3774

Delta costi dovuto principalmente alla riduzione del tempo di impiego delle risorse non appartenenti al team di sviluppo, incaricate della gestione di tutte le fasi ad esclusione di quelle di test e sviluppo. Tali fasi sono solitamente gestite da risorse interne all'azienda, che non saranno più impiegate, grazie alla notevole riduzione degli stati del workflow e direttamente amministrare dal team di sviluppo. Risorse interne aventi un costo medio orario inferiore rispetto al precedente pari a circa 13 euro all'ora. Perciò il beneficio di costo dovuto al non impiego di tali risorse, potrà essere calcolato come il prodotto del delta tempo risparmiato in ore per il costo medio orario della singola risorsa interna.

5.3 Qualità

In ottica di valutazione della qualità, potremo definire due KPI, la cui valutazione effettiva potrà però essere unicamente valutata mediante l'applicazione reale del progetto nella modalità agile:

Nel dettaglio potremo definire:

1. Efficienza

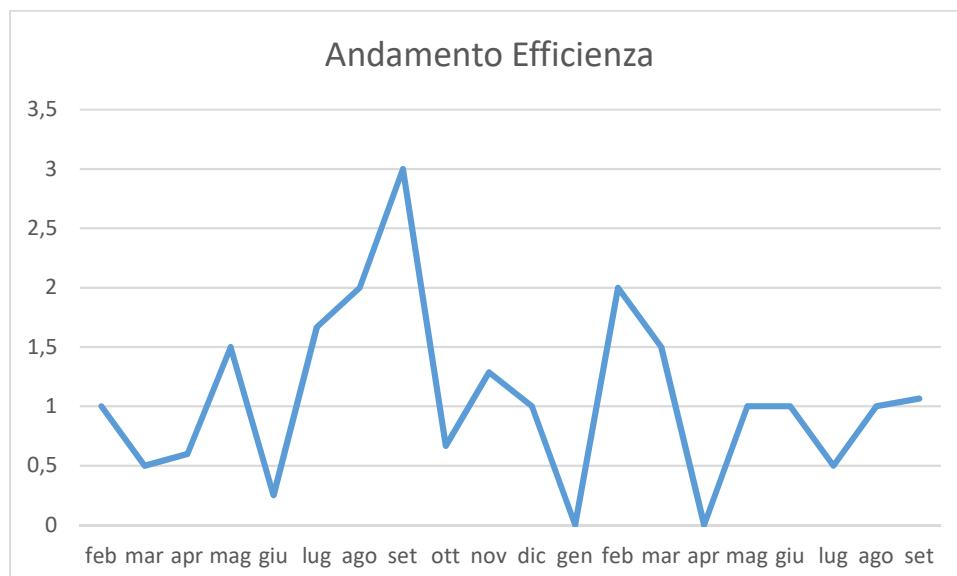
L'analisi si è focalizzata sulla efficienza del team nella risoluzione delle richieste e la sua velocity. Indicatore di performance che mira a valutare la “capacità costante di rendimento e di rispondenza alle proprie funzioni o ai propri fini nella pienezza delle proprie capacità o possibilità” (treccani.it), al fine di valutarne il bilanciamento del carico di lavoro assegnato al team nel tempo e la periodicità con cui quest'ultimo rilascia valore per l'azienda (inteso come richieste chiuse, associate a nuovi sviluppi richiesti dall'utente)

Nel dettaglio il seguente KPI è stato valutato prendendo in considerazione il numero di richieste chiuse su quelle aperte mensilmente:

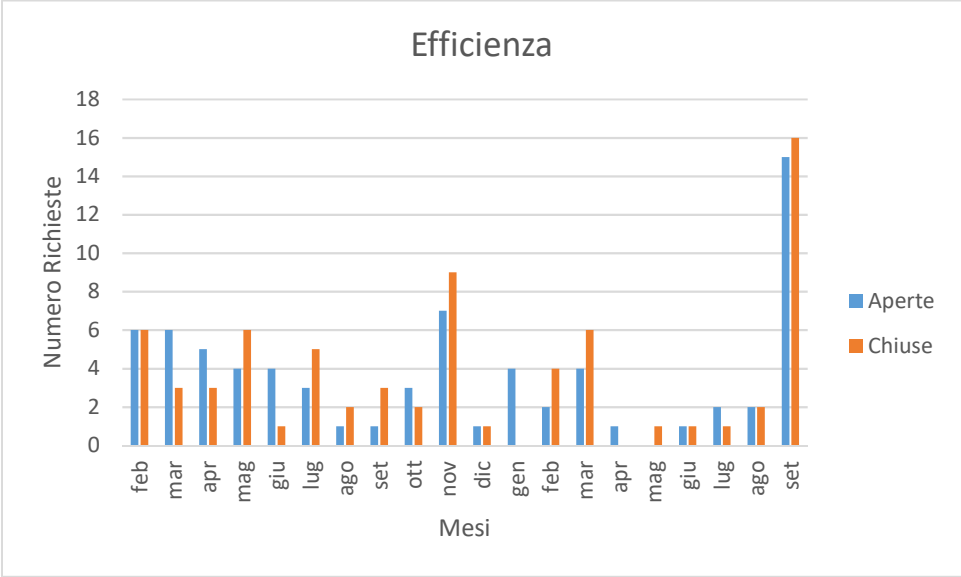
$$0 < Efficienza = \frac{Richieste\ Chiuse}{Richieste\ Aperte}$$

L'indicatore risulterà tanto più ottimale quanto più si allontanerà dallo zero.

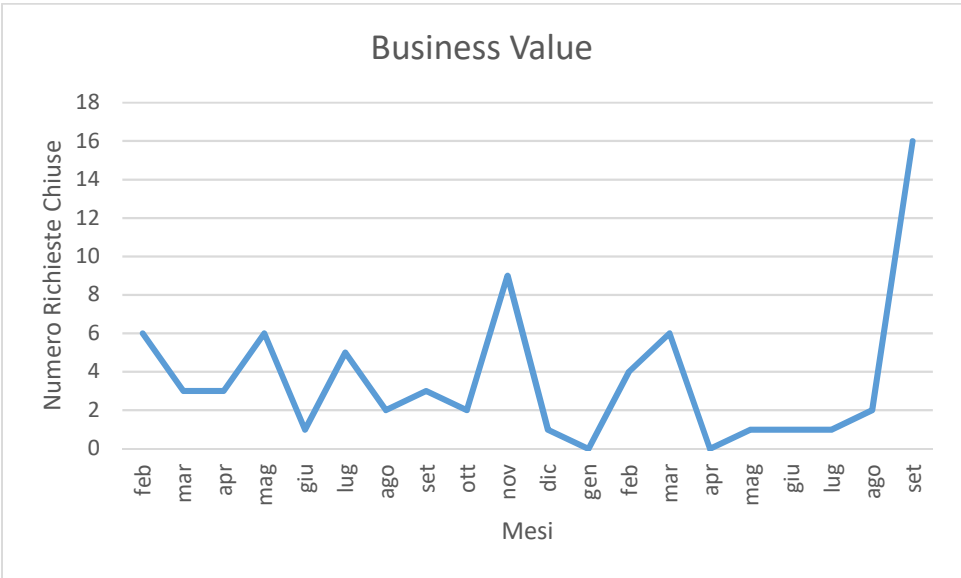
A seguito dell'analisi si è potuto osservare che l'efficienza, nella situazione attuale, non risulta costante nel tempo, in quanto alterna mesi in cui l'indicatore è maggiore di 1, grazie alla chiusura di più richieste di quelle aperte, a mesi dove quest'ultimo risulta pari a 0.



La natura non costante dell'indicatore è principalmente dovuta alla durata prevista e stimata per le richieste non uniforme e bilanciata. Infatti le richieste con durata elevata, potranno essere aperte in un mese e chiuse solo a distanza di mesi. Nel grafico di seguito si può osservare nel dettaglio il numero di richieste aperte e chiuse effettivo mensilmente.



La conseguenza di una efficienza non bilanciata e costante sarà una distribuzione della capacity delle risorse di sviluppo altrettanto non uniforme e la possibilità di non poter deliverare in maniera continua valore per l'azienda ed il cliente, percepito unicamente a chiusura delle richieste, quindi a conclusione dello sviluppo e dell'implementazione di nuove caratteristiche.



Grazie alla metodologia agile l'indicatore potrà essere reso costante nel tempo e pari circa ad 1.

Il livellamento dell'efficienze sarà possibile, come già spiegato in precedenza, grazie alla suddivisione di tutte le attività in task con durata prevista minore e soprattutto uniforme.

In tal modo, dunque, sarà anche possibile produrre valore per l'azienda ed il cliente in maniera continua e costante con periodicità bisettimanale, pari alla durata definita per lo sprint, e bilanciare anche il carico di lavoro delle stesse risorse di sviluppo.

2. Service Level

La definizione del seguente KPI ha lo scopo di verificare uno dei capisaldi dell'agile, cioè la più facile e flessibile gestione dei problemi e bug riscontrati a seguito degli sviluppi e del progetto possibile grazie agli sprint con pianificazione ed esecuzione periodica che non obbligano il pm a dover ridefinire la pianificazione delle attività e al ritardo nell'esecuzione e completamento delle stesse.

L'indicatore di performance, coerentemente alla tipologia di richieste utilizzate nella gestione, sarà quindi così definito:

$$0 < \textit{Service Level} = \frac{\textit{Incident Risolti}}{\textit{Incident Aperte}} < 1$$

In particolare, nella gestione waterfall, tali problemi saranno definiti incident o bug, in quella agile invece, come micro attività di durata breve, direttamente gestiti e risolti nello sprint più prossimo.

Tuttavia il calcolo di tale KPI potrà avvenire solo in seguito all'incombere di problemi sul progetto dovuti ad eventuali errori di rilascio, ancora non presenti e riscontrati.

5.4 I Benefici del Knowledge management

Nel paragrafo si approfondiranno i benefici ottenuti dall'introduzione del un nuovo tool tecnologico per la gestione della conoscenza Confluence.

Come descritto nel capitolo precedente l'applicativo introdotto ha permesso una capitalizzazione del progetto e della conoscenza derivante in modo semplice, veloce ed integrato con il sistema di gestione dello stesso progetto Jira.

In particolare i benefici derivanti dall'introduzione del Knowledge management possono essere identificati qualitativamente in:

- a) Riduzione tempi di sviluppo: mediante la formalizzazione e standardizzazione delle procedure operative e di risoluzione all'interno dell'applicativo, viste in precedenza, facilmente ricercabili attraverso la ricerca intelligente fornita dallo stesso.
- b) Introduzione di una repository documentale: la possibilità di collezionare tutta la documentazione necessaria riferita ad un progetto all'interno di un ambiente unico condiviso con tutti gli attori dello stesso, a differenza della situazione attuale nel quale vi sono più cartelle condivise in rete suddivisa per gruppi di lavoro o progetti origine di confusione e disordine.
- c) Modelli standard unificati: l'introduzione di modelli e processi standard di scrittura documentale non creanti malintesi o libere interpretazione, evitanti quindi ritardi dovuti ad errate comprensioni e letture.
- d) Monitoraggio facilitato: possibilità di poter monitorare lo stato delle richieste all'interno di ogni sprint e l'esecuzione dello stesso mediante l'integrazione diretta di Confluence con jira, grazie alla possibilità di salvare su confluence i grafici di andamento formulati su Jira al termine di ogni sprint solamente con un click. In tal modo, si avrà l'occasione di poter correggere negli sprint futuri, in modo quasi immediato, gli errori di stima o gestione commessi negli sprint terminati secondo le valutazioni effettuate nello sprint review e retrospective.

Capitolo 6: Conclusioni

La modalità di gestione del progetto descritto nell'elaborato ha permesso di individuare le fasi in cui deve essere suddiviso ogni progetto e i relativi ruoli chiave (es. Scrum Master) della metodologia Agile.

Le leve di miglioramento riguardano: l'imprecisione nella stima delle attività, la complessità dei workflow seguiti dalle richieste, il tempo impiegato per le attività non legate allo sviluppo, l'utilizzo improprio del know-how.

È stato necessario effettuare cambiamenti strutturali e organizzativi, definire delle assunzioni relative al tempo di test e tempo di sviluppo di ogni richiesta, definire dei range ideali in cui la durata ed il tempo di test di ogni richiesta devono ricadere per essere considerati plausibili.

Seguendo tali assunzioni, la trasformazione del progetto si è sostanziata in una suddivisione delle richieste in attività più piccole, schedulazione all'interno degli sprint bisettimanali definiti, oltre che la definizione del numero totale degli sprint necessari al completamento del progetto e alla sua durata.

Il "TO BE" è oltremodo caratterizzato dall'introduzione del nuovo sistema informatico per la gestione della conoscenza "Confluence". Questo ha permesso di diffondere il know how, un nuovo modo di miscelazione della conoscenza: uniforme, unico, non fraintendibile e sicuro.

A conclusione dell'esplicitazione della situazione agile è stato possibile effettuare una consequenziale analisi dei costi e benefici derivanti dal nuovo approccio.

Le aree d'analisi sono state: tempi, costi, qualità e conoscenza.

L'analisi costi - benefici pone l'attenzione sui tempi necessari alla gestione di ogni singola fase in cui la richiesta transita, che definirà la durata del progetto.

In primis si è analizzato proprio la durata del progetto, che è stato possibile completare mediante l'esecuzione di venticinque sprint, permettendone la chiusura nell'arco di un anno riscontrando un beneficio di tempo pari a 226 giorni.

Nel dettaglio, si è analizzato il tempo dedicato per i test, una delle fasi più critiche, per il quale, grazie alle assunzioni definite e le variazioni attuate con l'agile, si è osservato un beneficio temporale di circa 1893 ore, 47%, ed il tempo complessivo dedicato alle fasi restanti, ad

esclusione del tempo di sviluppo, per le quali si è constatato una riduzione di circa 990h pari al 77%.

Attenzione particolare è stata posta sulla fase di STIMA per la quale si è calcolato un indicatore di performance legato alla precisione della stessa, alla corrispondenza tra il tempo di sviluppo ipotizzato e quello effettivo: nella situazione AS-IS è risultata, nell'intorno da zero ad uno, in media dello 0.83. Grazie alla nuova scomposizione e gestione delle attività potrà tendere ad uno.

Nel secondo caso, si sono valutati i vantaggi ottenuti in ottica di costi, direttamente proporzionali a quelli dei tempi: minor utilizzo delle risorse, interne ed esterne che ha permesso di ridurre i costi mediamente del 47%.

Si sono definiti degli indicatori di qualità legati principalmente a due aspetti: efficienza e service level.

Il primo è stato definito come la capacità costante di rendimento e di rispondenza alle proprie funzioni o ai propri fini nella pienezza delle proprie capacità o possibilità, calcolato come l'ammontare delle richieste chiuse su quelle aperte mensilmente, il cui andamento nel tempo è risultato non costante e molto variabile.

Il secondo invece è stato definito come il livello di servizio della banca nella risoluzione degli Incident, quindi problemi causanti inoperatività, calcolato come il numero di incident chiusi su quelli aperti mensilmente, attualmente non calcolabile, in quanto non si sono ancora verificati all'interno del progetto.

Per quanto riguarda i benefici/costi dovuti alla conoscenza, o meglio grazie alla gestione di quest'ultima in maniera più strutturata, mediante l'utilizzo di Confluence, si sono riscontrati diversi benefici qualitativi e pertanto immateriali: l'introduzione di una repository documentale unica per tutte le business units bancarie, un monitoraggio delle attività più facilitato e una gestione della conoscenza standardizzata e unificata.

La dimostrazione dei benefici e degli indicatori di performance sarà possibile quantificarla, in ambito aziendale, quando la modalità Agile sarà adottata per tutti i progetti di dimensioni o caratteristiche simili a quello studiato, potendone effettivamente acquisirne benefici e costi sull'intero portfolio di progetti.

Bibliografia

- https://www.itconsult.it/knowledgebox/whitaper/PDFitc_WP_Formazione_Cap2.pdf
- *Scrum Body of Knowledge*
- *La Guida a Scrum™*
- <http://www.treccani.it>
- <https://confluence.atlassian.com>
- Abhinav Gupta, “*Value Proposition of Agile Development*”, January 2018
- <https://www.knowledgehut.com/blog/agile/value-proposition-of-agile-development>
- Acebes, Fernando, et al. “*A new approach for project control under uncertainty. Going back to the basics.*” *International Journal of Project Management* 32.3 :423-434, (2014)
- Ahlemann, Frederik. “*Towards a conceptual reference model for project management information systems.*” *International Journal of Project Management* 27.1 :19-30, (2009).
- Ahmad, Norita, and Phillip A. Laplante. “*Software project management tools: making a practical decision using AHP.*” *Software Engineering Workshop, 2006. SEW'06. 30th Annual IEEE/NASA. IEEE, (2006).*
- Aldea, Cosmina Carmen, and Cristian Olariu. “*Selecting the optimal software solution under conditions of uncertainty.*” *Procedia-Social and behavioral sciences* 109 333-337,(2014).
- Alshawi, Mustafa, and Bingunath Ingirige. “*Web-enabled project management: an emerging paradigm in construction.*”
- *Automation in construction* 12.4: 349-364, (2003).
- Archibald, Russell D. “*Managing high-technology programs and projects*”. John Wiley & Sons, (2003).
- Artto K., Martinsuo M., Kujala J., *Project business*, Helsinki, (2011).
- B. Boehm. *Get ready for agile methods, with care. Computer*, 35(1):64–69, (2002).
- Bandor, Michael S., “*Quantitative methods for software selection and evaluation.*” No.
- CMU/SEI-2006-TN-026. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, (2006).
- Beck, Kent, et al. “*Manifesto for agile software development.*” (2001): 2006.
- Beck, Kent, and Erich Gamma. “*Extreme programming explained: embrace change.*” addison-wesley professional, (2000).77

- Bigatti Massimiliano, *“Extreme Programming e Metodologie Agili di Sviluppo Software: concetti, prodotti e risorse”*. (2002)
- Braglia, Marcello, and Marco Frosolini. *“An integrated approach to implement project management information systems within the extended enterprise.”* International Journal of Project Management 32.1: 18-29, (2014).
- Caniëls, Marjolein CJ, and Ralph JJM Bakens. *“The effects of Project Management Information Systems on decision making in a multi project environment.”* International Journal of Project Management 30.2: 162-175, (2012).
- Cantamessa, Marco, and Francesca Montagna. *“Management of innovation and product development”*. London: Springer, (2016).
- Casoni Mirco, *“Scrum: un processo agile”*, September 2010,
- <http://www.mokabyte.it/2010/09/scrum-1/>.
- Cohn, Mike. *“User stories applied: For agile software development.”* Addison-Wesley Professional, (2004).
- Coram, Michael, and Shawn Bohner. *“The impact of agile methods on software project management.”* Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE
- International Conference and Workshops on the. IEEE, (2005)
- De Marco, *“Project Management Theory Book”* URL:<http://hdl.handle.net/11583/1659508>, (2007).
- E. Arisholm, H. Gallis, T. Dybå, and D.I.K. Sjøberg. *“Evaluating pair programming with respect to system complexity and programmer expertise. IEEE Transactions on Software Engineering”*, 33(2):65–86, (2007).
- Eldrandaly, Khalid. *“GIS software selection: a multicriteria decision making approach.”* Applied GIS 3.5: 1-17, (2007).
- Górecka, Dorota. *“Multi-Criteria Decision Aiding in Project Management-Outranking Approach and Verbal Decision Analysis.”* Studia Ekonomiczne 137: 11-38, (2013).78
- Guide, A. *“Project Management Body of Knowledge (PMBOK® GUIDE).”* Project Management Institute. (2001).
- Harter Jim, *“Why managers are central to an agile culture.”* Gallup, October 2018,
- <https://www.gallup.com/workplace/243455/why-managers-central-agile-culture.aspx>
- Hazır, Öncü. *“A review of analytical models, approaches and decision support tools in project monitoring and control.”* International Journal of Project Management 33.4: 808-815, (2015).

- Hu, Zhi-gen, Quan Yuan, and Xi Zhang. "Research on agile project management with scrum method." Services Science, Management and Engineering, 2009. SSME'09. IITA International Conference on. IEEE, (2009).
- J. McAvoy and T. Butler. "The role of project management in ineffective decision making within agile software development projects." European Journal of Information Systems,18(4):372–383, (2009).
- Kaygısız-Ertuğ, Z., & Girginer, N. "A multi criteria approach for statistical software selection in education." Hacettepe Üniversitesi Eğitim Fakültesi Dergisi [Hacettepe University Journal of Education], 29(2), 129-143, (2014).
- Kasra Madadipouya "An Examination and Evaluation of Agile Methodologies for Systems Development." Australasian Journal of Computer Science, 2: 1-17, , (2015).
- Kerzner, Harold, and Harold R. Kerzner. "Project management: a systems approach to planning, scheduling, and controlling". John Wiley & Sons, (2017).
- Kittlaus, Hans-Bernd, and Peter N. Clough. "Software product management and pricing:Key success factors for software organizations." Springer Science & Business Media,(2008).
- Love, Peter ED, et al. "The enigma of evaluation: benefits, costs and risks of IT in Australian small–medium-sized enterprises." Information & Management 42.7: 947-964 (2005).
- Mark, A. "PMI's Pulse of the Profession: The High Cost of Low Performance. A Core Competency for Project and Program Success." Project Management Institute, (2014).79
- Pressman, Roger S., and Elena Addomine. "Principi di ingegneria del software."McGraw-Hill, (1991).
- Raymond, Louis, and François Bergeron. "Project management information systems: An empirical study of their impact on project managers and project success." International Journal of Project Management 26.2: 213-220, (2008).
- S. Ambler. "Agile software development at scale. Balancing Agility and Formalism in Software Engineering", pages 1–12, (2008).
- Saaty, T.L. "Fundamentals of the Analytic Hierarchy Process," RWS Publications, Pittsburgh, PA, (2000).
- Sakka, Ouafa, Henri Barki, and Louise Côté. "Relationship between the interactive use of control systems and the project performance: The moderating effect of uncertainty and equivocality." International Journal of Project Management 34.3: 508-522, (2016).
- Serrador, Pedro, and Jeffrey K. Pinto. "Does Agile work? A quantitative analysis of agile project success." International Journal of Project Management 33.5 :1040-1051, (2015).

- Shim, Jung P., et al. *"Past, present, and future of decision support technology."* Decision support systems 33.2: 111-126, (2002).
- Sutherland, Jeff, et al. *"The scrum papers: Nuts, bolts, and origins of an agile process."* (2007).
- Tarantilis, Christos D., Chris T. Kiranoudis, and N. D. Theodorakopoulos. *"A Web-based ERP system for business services and supply chain management: Application to real world process scheduling."* European Journal of Operational Research 187.3: 1310-1326,(2008).