GESTIONE MANAGERIALE NELPROCESSO DI FEATURE DELIVERY DI SISTEMA ODI SOFTWARE IN AMBITO MECCANICO, AERONAUTICO ED AUTOMOBILISTICO METODOLOGIE WATERFALL ED AGILE A CONFRONTO

EMANUELE BASILE
CON LA COLLABORAZIONE DI MICHELE NAVA E GIOVANNI NICOLAI

INTRODUZIONE

Dispositivi complessi di nuova generazione utilizzabili in campo meccanico, automobilistico o aeronautico stanno diventando simili aapparecchiature IT. Sempre più spesso, infatti, si creano sistemi combinati in cui l'architettura di sistema o il software è asservito, o controlla, sistemi meccanici. Per questo motivo, lo sviluppo e la realizzazione di alcune parti di suddetteattrezzature possono richiedere soluzioni alternative ed innovativedi derivazione anche diversa da quella tradizionalmente usata in campo meccanico. In parallelo alle nuove esigenze di mercato, alcune tecniche di project management per queste tipologie di progetti legati ad apparecchi ibridi, ovvero ottenuti per combinazione di componentistica meccanica (automotive ed aeromotive) e componentistica IT, possono essere ispirate a settori non strettamente relati all'industria tradizionale ma ad ambiti differenti, quali quellodella tecnologia dell'informazione o le software house. Nell'ultimo decennio, l'industria meccanica ha visto un crescente servirsi (ed asservirsi) di metodologie di controllo a calcolo numerico più che richiedere un diretto coinvolgimento dell'operatore. Per alcune funzionalità di sistemi anche vitali e complessi, è stato rilevato un costante aumento di risorse informatiche in settori tradizionalmente orientati ad un ambito meccanico. Questa nuova domanda richiede un numero crescente di risorse dedicate allo sviluppo di software e firmware che ospitano le "linee di comando" della gestione del componente (generalmente note come "caratteristiche" o "feature"). Sono pertanto possibili analogie tra le software house di aziende operanti in ambito meccanico in generale, con quelle di firme operanti in ambiti strettamente informatici. In ambito automobilistico ed aeronautico si stanno diffondendo sempre più rapidamente piattaforme unificate in grado di gestire differenti funzionalità di un veicolo o un aeromobileche vanno dalla gestione interfaccia con l'utente al controllo di pilotaggio o guida autonoma (per il settore automotive questa funzionalità è conosciuta come ADAS - Advanced Driver Assistance System). Queste feature includono la gestione ottimizzata del sistema di consumo (fuelefficency) o il controllo della stabilità del velivolo (o veicolo) anche in differenti condizioni atmosferiche o stradali a seconda che si faccia riferimento ad un aeromobile o ad una automobile. Data la complessità di questi sistemi di controllo integrato, si stanno creando collaborazioni tra aziende operanti nello stesso settore o in settori similari miranti allo sviluppo di sistemi di controllo IT comuni a più firme ed a più sistemi. Questo approccio unificato introduce un'importante unificazione, quindi un risparmio sui costi, nell'intera linea produttiva. D'altra parteil suddetto modus operandi, porta inevitabilmente complessità di implementazione e richiede risorse più altamente qualificate. Riduzione costi, rapidi cambiamenti nella definizione dei requisiti, flessibilità nell'attuazione e soddisfazione del cliente, implicano un processo di implementazione diverso non solo dal punto di vista tecnico ma anche dal lato della gestione progetto.

GENERALITÀ

Le attuali tecniche di sviluppoutilizzate in ambito meccanico aeronautico e automobilistico, principalmente basate su set di requisiti definiti interamente nella fase iniziale del processo (metodologia Waterfall),sono qui presentate e confrontate con un approccio alternativo per l'implementazione delle funzionalità in cui i requisiti sono definiti in fasi sequenziali in base al feedback degli stakeholder (approccio Agile). Le due metodologie sono alternative, ovvero possono essere utilizzate entrambe a seconda delle scelte aziendali o dei gruppi di lavoro coinvolti. Tuttavia ci sono casi in cui il primo approccio non è praticamente rimpiazzabile con il secondo mentre in altre circostanze le due metodologie sono alternative e la scelta tra

le due è di natura economica o relata alle strutture organizzative dei team di lavoro. In generale, la metodologia Agile ha una buona applicabilità in quei contesti in cui non si ha unoscopo di progetto completamente definito nella fase iniziale ed è necessaria una maggiore sperimentazione e definizione nelle fasi successive del progetto. La metodologia Agile infatti funziona principalmente quando i requisiti iniziali e le tecnologie applicate hanno un range medio di incertezza. Progetti in cui si ha elevata incertezza sia a livello di tecnologia da utilizzare che di requisiti iniziali, non permettono un buon utilizzo di nessuna delle due metodologie. In modo analogo, laddove il progetto richieda alti investimenti in fase iniziale ed abbia un alto livello di confidenza nella definizione dei requisiti sin dalle prime fasi di definizione, si potrebbe preferire un adozione di un metodo più tradizionale (Waterfall).

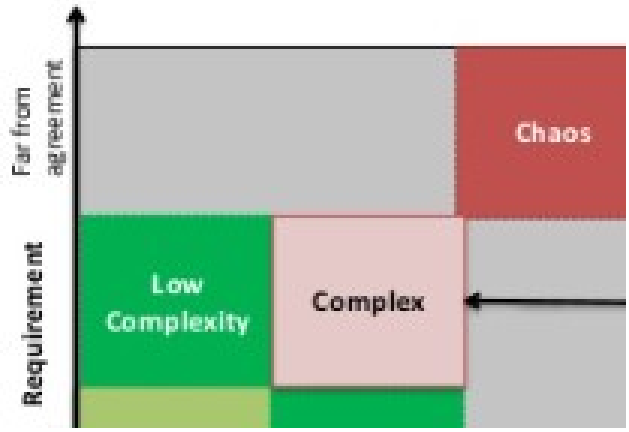

Figura 1 – Scenari in cui la metodologia Agile risulta essere maggiormente efficace

DESCRIZIONE DEL PROCESSO – STATO ATTUALE

Secondo la stato dell'arte per buona parte dei progetti in ambito meccanico, l'implementazione inizialmenteprevede una definizione della funzionalità ad alto livello per arrivare ad una più dettagliata descrizione della stessa prima di iniziare la fase esecutiva, Rif. [1]. In parallelo all'esecuzione da parte del project management si esegue un processo di monitoraggio e controllo per allineare le aspettative di progetto con lo stato reale e assicurare che impedimenti, quali rischi oblocchi, vengano rimossi o, per lo meno, si mettano in atto adeguate strategie per limitare la loro probabilità ed il loro impatto sul progetto. A seguito della fase esecutiva, quando il cliente ha accettato le consegne (delivery), segue una fase di chiusura progetto e rilascio dei team che hanno lavorato su di esso. La Figura 2 mostra un processo di creazione e realizzazione di una generica feature secondo un processo tradizionale (Waterfall) ad oggi in uso in industrie meccaniche.Una nuova feature o un cluster di nuove funzionalità sono create a partire da una richiesta del mercato, determinata mediante benchmarking, sondaggi sui clienti, vantaggio competitivo o esigenza normativa. Il processo inizia con una definizione di massima della feature a partire dalle esigenze di progetto e termina con un'implementazione delle funzionalità nell'architettura di sistema prima di essere testato e quindi messo sul mercato.

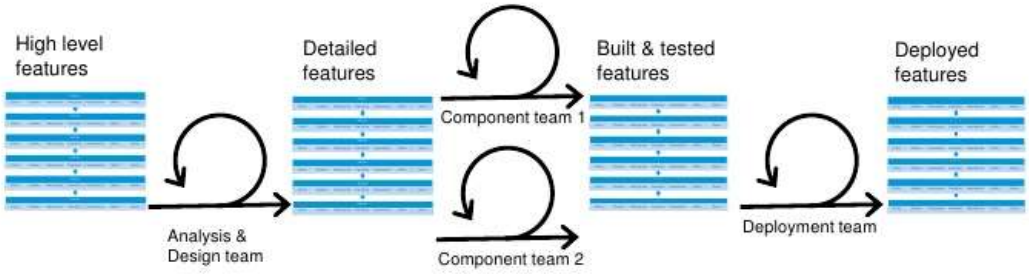

Figura 2 – Feature delivery, panoramica generale

Con riferimento ad un processo di implementazione di feature attualmente adottato in differenti contesti, il presente studio fa riferimento ad una nota azienda automobilistica europea, la metodologia Waterfall prevede l'esecuzione di quattro fasi sequenziali. Nella definizione dei requisiti del cliente (o SC – System Concept), viene presentata una descrizione generale della funzione e viene assegnato un budget adeguato a seconda delle diverse divisioni coinvolte nell'implementazione della feature. Questa valutazione si basa su stime preliminari della complessità dell'attività e delle risorse necessarie. La fase seguente è la definizione dei requisiti di sistema (o SD - System Design) in cui ogni aspetto della funzionalità viene analizzato e confrontato con le normative vigenti. In questa fase viene creata una seconda stima dell'impatto delle funzionalità e, se necessario, viene assegnato ulteriore budget. La progettazione del sistema è una fase importante del processo di implementazione, che termina con la decisione "go / no go" di implementare una funzionalità nell'architettura di sistema. La fase successiva è System Architecture Definition (o SA - System Architecture) in cui l'obiettivo è la definizione della funzionalità secondo una specifica architettura. In questa fase, le nuove funzionalità sono incluse nell'architettura preesistente assicurandosi che non vi siano conflitti o incompatibilità. Viene eseguito un controllo finale prima di consegnare la funzione (che ora è denominata ticket) ai requisiti del sottosistema e alla definizione dell'architettura (o NR - Network Release) di Figura 3 –.

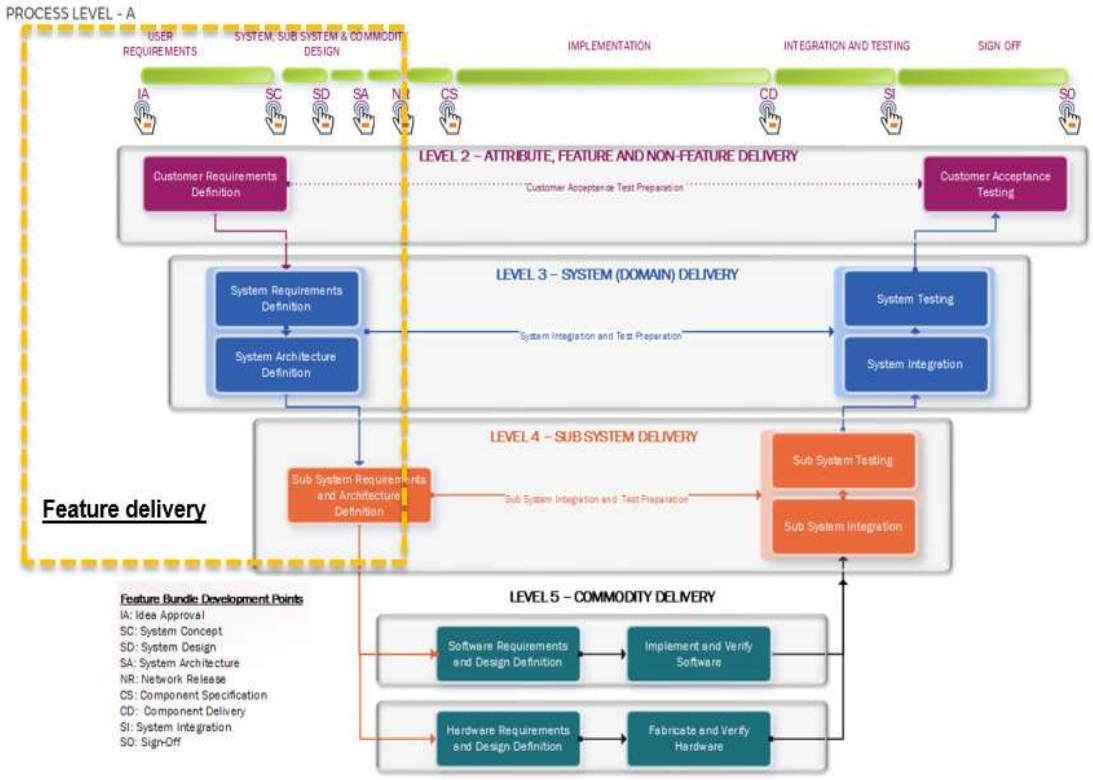

Figura 3 – Processo di feature delivery

Queste fasi presentate sono implementate in momenti sequenziali. Una serie di funzionalità (di solito circa 20) vengono implementate ogni 3 mesi nelle fasi SC e SD. Una volta completate, queste feature vengono trasferite al team SA che implementa quanto completato in SD ed altri ticket dovuti a revisioni interne. La quantità di ticket che viene implementata ogni tre mesi (intervallo di iterazione) nello spazio SA è di circa 70 unità. Al termine dell'implementazione SA, il nuovo pacchetto di funzionalità viene trasferito al NR che completa l'implementazione nel mese successivo. La consegna in rete rilascia un nuovo software in un lasso di tempo fisso e non modificabile, lo scopo del progetto, tuttavia, può cambiare (de-scoping) per garantire che ogni consegna avvenga nei tempi concordati. Tutte queste fasi sono sequenziali ed implementate in un approccio con metodologia Waterfall, ogni fase termina con un gateway che consente l'avvio della fase successiva. Se un ticket non viene completato entro i tre mesi di esecuzione, passa alla successiva iterazione e verrà consegnato nel successivo pacchetto di funzionalità. Eccezioni sono trattate su base individuale e richiedono una ufficiale approvazione in una Change Control Board (CCB). Il processo analizzato è mostrato inFigura 3 – e fa parte del sistema di implementazione che, assieme ad integrazione e test, completa l'intero processo di realizzazione e consegna al cliente di una feature in un'architettura di sistema.

Per garantire che una feature venga consegnata in tempo per uno specifico modello (Model Year) ma anche una fase specifica dell'assemblaggio del componente, la tempistica deve essere concordata in anticipo con ciascun proprietario della funzione.

In sintesi, un approccio di tipo Waterfall è tradizionale per il settore automotive ed è caratterizzato da "un rigoroso processo sequenziale di consegna di prodotti o servizi", Rif. [2]. Ogni iterazione (o bundle) è composta da:

- Ricerche di mercato condotte per set completi di funzionalità nella fase di definizione concetto,SC (3 mesi secondo il processo di consegna delle funzionalità sopra descritto),
- Implementazione della progettazione del sistema delle funzionalità che comprende un'indagine completa e una descrizione dei requisiti applicabili, fase SD (3 mesi),
- Implementazione dell'intero set nell'architettura di sistema, fase SA (3 mesi),
- Sviluppo e rilascio della rete, fase NR (1 mese)
- Test finali e correzione dei bug

Per la metodologia Waterfall, si richiede grande attenzione nell'avere requisiti completi, ben definiti e un piano di progetto dettagliato sin dalle prime fasi di progetto, Rif. [3]. La Tabella 1 riassume pro e contro di questo approccio.

PROs	• Ogni fase del processo è completamente documentata e può essere gestita e controllata in dettaglio dalle prime fasi di pianificazione progetto alla sua conclusione • L'intero processo di consegna è lineare e facile da monitorare • Buona allocazione delle risorse e controllo dei costi durante tutte le fasi
CONs	• Bassa flessibilità nell'accomodare nuove richieste specialmente nelle ultime fasi di implementazione • Ogni feature richiede un stesso lasso temporale per essere completata (indipendentemente dalla sua complessità) • Modifiche alla feature incluse in una delivery, dovute ad esempio ad obsolescenza, sono difficili da gestire una volta iniziatala fase di esecuzione

Tabella 1 - PROs and CONs della metodologia Waterfall per il processo di feature delivery

DESCRIZIONE DEL AGILE

Un approccio alternativo (basato su metodologia Agile), Rif. [4]) può essere utilizzato per implementare funzionalità nell'intero processo di feature delivery. Questa metodologia si basa su definizioni di requisiti più flessibili inseribilicon tempistiche diverse nella fase di delivery delle feature. In questa maniera, le featurehanno una maggiore continuità di implementazione attraverso le varie fasi dalla definizione a livello di concetto (fase SC) alla consegna della feature (o ticket) nell'architettura di sistema (completamento della fase NR). Questo approccio, inoltre, rimuove il vincolo di avere un periodo di consegna fisso per quelle feature che possono essere ultimate in tempi più brevi rispetto a quelli dettati dalla durate della iterazione. Inoltre, questa metodologia funziona anche per quelle feature che sono in ritardo nel processo o per quei ticket che richiedono periodi di implementazione maggiori di una singola iterazione. Infatti, tale processo si basa sull'avere ogni feature, anche complessa, definibile come combinazione di più feature di dimensioni minori (processo di slicing) e quindi implementabili nel timeframe di una interazione singola. Secondo questo criterio pertanto, ogni feature, per quanto complessa,può essere implementata come successive ed indipendenti funzionalità da consegnare in cascata in diverse interazioni sino ad ottenere la funzionalità complessiva desiderata. Avere un approccio più flessibile consente un migliore allineamento tra funzionalità implementate e requisiti aggiornati [4]. Quindi, se una funzione richiede un cambiamento nel suo sviluppo, può essere facilmente adattata senza richiesta formale di modifica (processo di CCB).

Per rendere questo processo più efficiente, invece di considerare la singola iterazione come una entità unica, essa può essere divisa in diversi sprint di diversa lunghezza a seconda della fase. Con riferimento all'esempio presentato in precedenza del processo di implementazione feature per aziende operanti nel settore automotive, uno sprint di due settimane si adatta alla fase di implementazione nel network (NR), mentre uno sprint di 30 giorni potrebbe funzionare meglio per l'architettura di sistema (SA), per la fase di progettazione del sistema (SD) e per quella dello sviluppo del concetto (SC). La scelta di avere una lunghezza diversa dipende dal numero di feature da trattare in ogni fase, dalla loro complessità e da come i diversi stakeholder rispondono alla sua implementazioni in termini di come loro supportano il processo. Ogni sprint inizia e termina in un determinato e non modificabile periodo di tempo, tuttavia le attività che non vengono completate nel periodo assegnato vengono automaticamente assegnate al successivo, questa operazione è definita con la denominazione di "re-priorizzazione del backlog" e deve essere eseguita con regolarità durante tutta la fase di esecuzione con lo scopo di eliminare quelle funzionalità non richieste e priorizzare quelle che invece aggiungono maggior valore al prodotto finale. L'ultimo sprint di ogni fase rappresenta un'eccezione; se un ticket di questo sprint non viene completato entro il periodo di tempo assegnato, può essere assegnato alla successiva interazione oppure può essere rivisitato, ridotto e quindi completato con un contenuto limitato nella corrente. Questo approccio consente una migliore distribuzione del carico durante l'intera interazione ed evita una aumento del volume delle attività nell'ultima parte di essa. Inoltre, l'avere un approccio Agile nell'intero processo di feature delivery crea una migliore soddisfazione del cliente grazie a una definizione dei requisiti più flessibile. Feature aventi una maggior facilità di implementazione possono essere inserite in un arco di tempo più veloce e possono essere completate prima. Allo stesso modo, se una feature è ben supportata durante l'intera fase di implementazione, può essere completata in un tempo più breve nel network di sistema.D'altro canto se una funzionalità perde supporto può essere parcheggiata in attesa che le parti interessate siano più coinvolte senza vincolare l'intero pacchetto di consegna. In questo modo si aumenta l'attenzione nell'avere una stretta collaborazione tra sviluppatori di software e proprietari di feature nell'intero processo di implementazione. Avere una definizione di requisiti minimi, invece di una serie completa di requisiti secondo l'approccio Waterfall, riduce al minimo il tempo impiegato nella preparazione delle funzionalità aumentando il focus sulle attività che realmente contano per la consegna richiesta. Si ha in questo modo un immediato ritorno di investimento specie se si priorizzano in backlog quelle feature con maggior valore aggiunto. Le discussioni faccia a faccia sono maggiormente promosse al fine di mantenere un allineamento

costante tra gli sviluppatori e i team proprietari delle feature. In questo modo, qualsiasi mancanza di definizione può essere chiarita e possono essere creati requisiti più robusti durante la fase di esecuzione (riduzione del cosiddetto “golfo di valutazione”, ovvero il gap esistente tra le aspettative del cliente e le sue vere necessità). D’altro canto, la mancanza di definizione completa in fase iniziale può essere vista come un aspetto negativo dell’approccio Agile perché può portare ad attività aggiuntive per arrivare alla definizione della feature richiesta dal cliente, si richiede pertanto un processo di monitoraggio e controllo più attento per coordinare queste attività, Rif.[5]. La Tabella 2 riassume i pro e contro dell'approccio Agile.

PROs	• La definizione e l'implementazione delle funzionalità èeseguita in tempi brevi • Il cambiamento dei requisiti può essere accettato prima (priorizzazione del backlog) • Stretta collaborazione tra sviluppatori di sistemi e proprietari delle feature • Focus maggiore sullo sviluppo di feature • Miglioramento continuo (Kaizen), Rif. [4] • Maggiore controllo delle risorse
CONs	• Feature più complesse potrebbero essere penalizzate a causa dell'aumento del numero di team coinvolti nella loro implementazione • Le featurenon aventi un buon livello di coinvolgimento da parte degli stakeholder possono essere ritardate o parcheggiate (anche se altamente richieste dal cliente)

Tabella 2 - PROs and CONs della metodologia Agile, feature delivery

APPROCCIO AGILE IN ALTRI SETTORI

L'origine della metodologia Agile è da trovarsi in quelle attività che richiedono più know-how e meno investimenti in asset iniziali. Infatti,in quelle aree in cui vi è un elevato livello di incertezza nella definizione iniziale, le attività correlate sono meno definibili rispetto a un flusso di lavoro industriale più tradizionale, Rif. [7]. Un processo completamente definito, basato sulla tecnica Waterfall, rimane un percorso preferito per quei processi che potrebbero essere completamente definiti nelle fasi iniziali soprattutto se è richiesto un importante iniziale investimento. D'altra parte, le soluzioni iterative,utilizzate in Agile, possono essere adottate in quelle aree in cui le fasi sequenziali sono più dipendenti dalle fasi precedenti e dai risultati sperimentali nonché dai riscontri dei vari stakeholder. Per quei processi che non richiedono investimenti di partenzaimportanti e le attività possono richiedere modifiche in corso d’opera senza influire pesantemente sui costi iniziali, un'implementazione più flessibile può essere considerata un approccio valido come sostituto ad una metodologia più rigorosa in termini di iniziale definizione. Il principio Agile di base, è quello di dividere l'intera produzione in piccole sottoinsiemi (sprint) che possono essere implementati rapidamente dando un rapido riscontro con il cliente. In questo modo, è possibile implementare modifiche riducendo i tempi di consegna. Diverse società di sviluppo software stanno attualmente utilizzando l'approccio Agile con buoni risultati. La metodologia è stata inoltre trasferita ad altri settori come quello aerospaziale, delle telecomunicazioni e delle costruzioni.

Facendo riferimento a un approccio più tradizionale (Waterfall), l'ambito dei requisiti viene definito all'avvio del progetto in maniera rigorosa. Tempistiche e costi devono assecondare il conseguimento dello scopo del progetto. Cambi di scopo richiedono accettazione formale d parte del cliente (ChangeRequest Board).

In contrapposizione a quanto detto per la metodologia Waterfall, lavorare in ambiente Agile consente di avere tempo e costi concordati e di variare lo scopo del progetto in base alla richiesta dei clienti, Rif. [7]. Il tradizionale triangolo di gestione del progetto (con scopo fisso, costi e tempi variabili)può essere

trasformato ed invertito per un processo Agile rendendo variabile lo scopo ma mantenendo costanti tempi e costi.

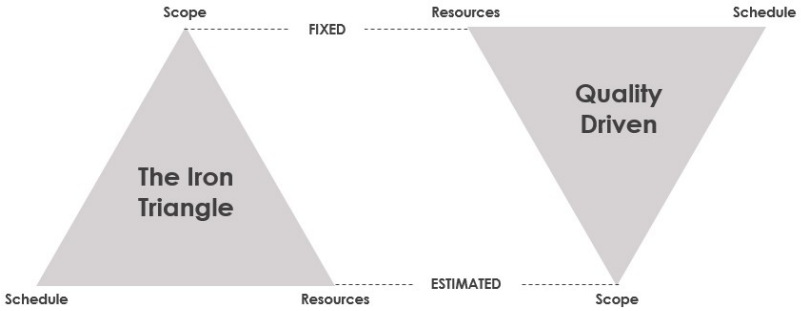

Figura 4 –Triangolo Waterfall (Iron) Vs Agile (qualitydriven), Rif. [8]

Esistono vari approcci Agile, quello presentato è basato su:

- Avere la completa trasparenza dell'intero risultato per garantire l'accordo delle parti interessate
- Controlli tempestivi dello stato del progetto e valutazione delle tappe intermedie
- Ridimensionamento del team in base alle esigenze del progetto

Le attività sono divise in vari sprint (o sotto-iterazioni) aventi lo scopo di creare un prodotto di valore minimo e sempre crescente da poter presentare al cliente per iniziale discussione. A seconda della fase, ogni sprint ha una lunghezza diversa per facilitare l'implementazione delle attività incluse, Figura 5. Ogni sprint può essere considerato un progetto a sé in cui qualsiasi modifica condiziona il risultato finale. I ridimensionamenti del team di solito non devono avvenire durante uno sprint ma nelle fasi finali. Le attività,che diventano obsolete durante l'esecuzione,devono essere rimosse dal deck di implementazione corrente, rivalutate o eliminate se non più necessarie, Rif. [9]. Ogni sprint inizia con un meeting iniziale in cui vengono stabilite le linee dettagliate di implementazione. Gli sprint terminano con un'analisi retrospettiva (retrospective) che consente ai membri del team di rivedere le fasi più importanti dell'esecuzione, evidenziare aspetti negativi e possibili soluzioni da implementare nelle successive esecuzioni. I retrospective rappresentano un buon modo per identificare regole e principi di base che potrebbero essere applicati a fasi o iterazioni successive per renderle più snelle ed efficienti (continuousimprovement o Kaizen, Rif. [4]). Un quotidiano meeting di start up, in cui il team presenta una panoramica delle attività correnti e crea un registro dei problemi, completa i set di riunioni relativealla definizione ed esecuzione di ogni sprint. Al termine di una iterazione, si eseguono test e verifiche di completezza attività con il cliente per garantire che il team e gli stakeholder siano d'accordo con la delivery appena completata.

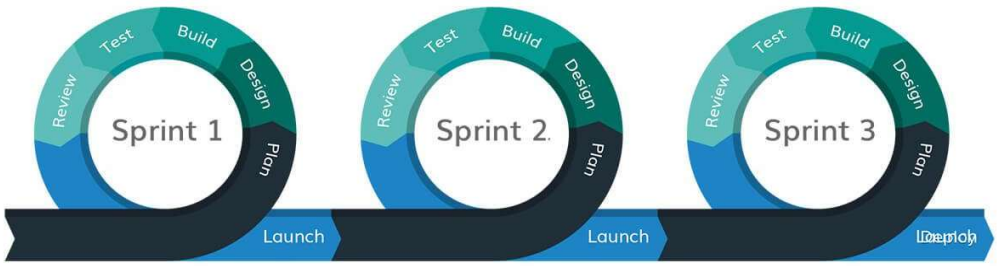

Figura 5 – Rif. [4] –approccio Agile in generale

L'idea di mantenere un backlog attivo durante ogni sprint consente di ridurre gli sprechi durante la fase di esecuzione e di mantenere il team concentrato sulle attività correnti. In ogni sprint le attività dovrebbero essere completate o parcheggiate perché, avendo attività che non stanno avanzando, porta ad una cattiva allocazione delle risorse. È necessario promuovere il multitasking e la riassegnazione delle attività in base alle competenze di maggior valore aggiunto per accelerare la consegna. Il team di implementazione deve essere completamente dedicato all'esecuzione dei task a loro assegnati. Il lavoro extra o aggiuntivo (goldplating) non è richiesto e pertanto deve essere evitato, solo il minimo livello di qualità accettabile dal cliente deve essere implementato e garantito durante le fasi esecutive. Un monitoraggio costante di attività collaterali deve essere condotto dai project manager e dai riferimenti tecnici al fine di avere una migliore distribuzione del carico di lavoro ed evitare creep di attività causato dallo sviluppare task non attesi o concordati con il cliente. Tecniche di priorizzazione di backlog sulla base della generazione di maggior valore aggiunto alla consegna o sulla precedenza di feature che richiedono maggiori tempi di implementazione, devono essere pianificate in anticipo come regole di base al fine di non creare situazioni di ambiguità durante la fase di esecuzione.

MODIFICHE USANDO L'APPROCCIO AGILE

L'approccio Agile può facilmente adattarsi al processo di feature delivery descritto in precedenza, esso può essere applicato alla totalità processo, partendo dalla definizione del concetto (in fase SC) sino al completamento della fase di consegna del nuovo network (in fase NR). Da notare come la metodologia Agile può essere adottata anche solo ad alcune delle fasi dell'intero processo ma, ovviamente, il metodo ha maggiore efficacia se tutta la linea di processo è allineata e permette l'implementazione in cascata di feature e ticket da una fase ad un'altra. In questa maniera, i differenti team implementano le 4 fasi presentate in Figura 2Figura 3 in modo sequenziale per ogni feature in modo individuale, ogni fase inizia quando la stessa feature viene completata nella fase precedente. In questa maniera ogni fase può essere condotta in parallelo alle altre e non in sequenza,le feature possono quindi passare da SC a SD, da SD a SA e quindi da SA a NR senza attendere la chiusura ufficiale di ogni fase.

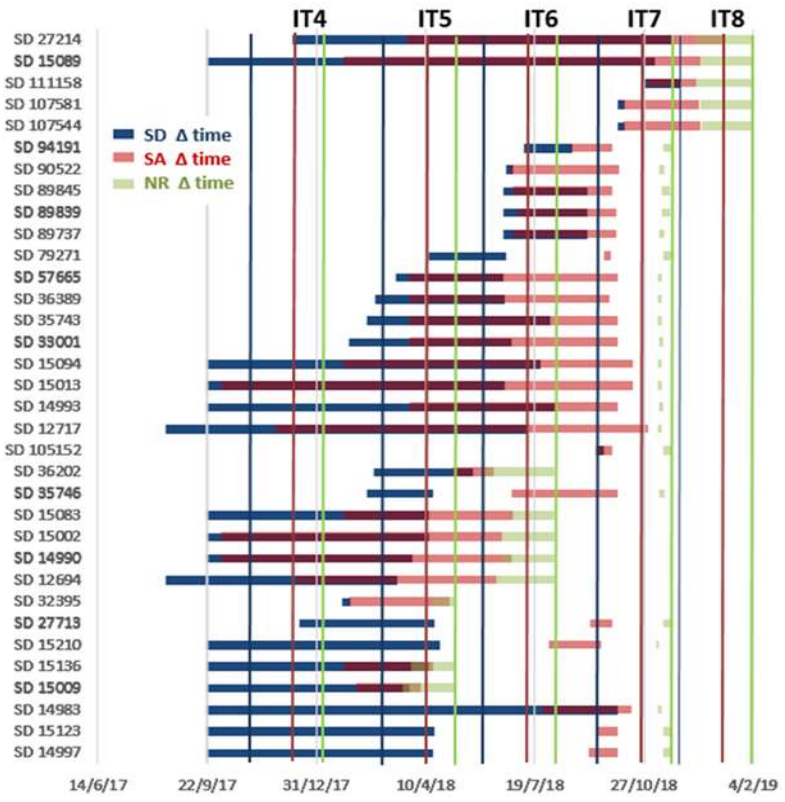

Figura 6 – Processo e durata implementazione delle feature

La Figura 6 riporta una rappresentazione grafica del processo secondo l'approccio attuale, ogni feature elencata sull'asse verticale è implementata in un approccio tipo Waterfall da una fase all'altra a partire da SD a NR (SC è stato omesso per semplificare la visualizzazione). Come si può vedere, alcune funzionalità richiedono più di un'iterazione per essere completate e si vanno a parcheggiare in SD o SA in attesa che il primo NR sia reso disponibile. Secondo l'attuale stato dell'arte, queste fasi sono non modificabili e, se una funzione non viene completata nel periodo di tempo concordato, deve essere spostata in una successiva iterazione. Ciò implica che le feature, anche se complete, attendono il completamento del processo di consegna senza poter essere processate nella fase successiva.

La Tabella 3 riassume alcuni importanti risultati. Ci sono alcune feature del processo che potrebbero essere completate in 16 giorni (6 in SD, 7 in SA, 3 in NR) ma richiedono 53 giorni in totale per essere pienamente implementate. La maggior parte di questo tempo (37 giorni) è tempo diattesa in cui la featureresta parcheggiata sino alla prima iterazione disponibile dopo il suo completamento. Utilizzando un approccio Agile alcune di queste funzionalità potrebbero essere implementate prima nel processo. Allo stesso modo, alcune funzionalità richiedono una lunga implementazione SD (su più di una iterazione) ma potrebbero essereimplementate in pochi giorni quando queste funzionalità sono in SA o NR. Ancora una volta, un approccio Agile può accelerarne il completamento e la featurepuò essere consegnata prima del completamente dell'intera fase SD nell'architettura (fase SA).

	System Design (SD)	System Architecture (SA)	Network Release (NR)	Total Length
Max Length (Days)	411	377	63	502
Average Length (Days)	160	135	21	270
Min Length (Days)	6	7	3	53
Standard Deviation	115	103	21	124
Number of Tickets	34	34	34	34

Tabella 3–tempi di implementazione media di risoluzione feature

Come detto, l'obiettivo di poter completare una funzionalità in un tempo più flessibile e di consegnarla non appena disponibile alla fase successiva (test) rappresenta uno stato cruciale del processo. Una piena convalida del processo si verifica solo quando viene testata l'architettura, i potenziali problemi potrebbero essere evidenziati prima se i test avvenissero con maggior tempismo e, se necessario, potrebbero essere rielaborati.

Un ulteriore miglioramento nella direzione di disporre di una soluzione più Agile in grado di consentire la consegna di funzionalità può essere quello di suddividere ciascuna feature in attività da implementare come sforzo combinato di ciascun team anziché come attività singola di un solo membro. La Figura 7 è un esempio di work load per una delle fasi della processo di feature delivery. Una struttura simile può essere adottata per il resto del processo rivalutando la lunghezza dello sprint, la complessità del ticket e le risorse presenti. Prima dell'inizio dell'attività in ogni sprint, è necessario predisporre alcuni attività di setting dello sprint per consentire un processo di implementazione più armonizzato. In questa fase (pianificazione), la disponibilità delle risorse deve essere valutata parallelamente alle attività da realizzare in uno sprint. Una seconda parte della pianificazione è condotta con il supporto di un esperto tecnico (o Scrum master) ed è correlata alla valutazione della complessità delle caratteristiche, all'identificazione consegnabile e alla definizione delle priorità. Le attività di gestione del progetto a supporto della fase di esecuzione saranno la gestione del rischio e della qualità, il controllo dei costi e delle risorse. Completano le attività di project management il daily start-up, per allineare la squadra e la definizione degli obbiettivi della giornata. Al termine della fase di esecuzione c'è la chiusura dell'interazione (retrospective) in cui vengono analizzati i

risultati e viene effettuata una valutazione della qualità per valutare gli aspetti tecnici dei risultati. Attività non complete devono essere trasferite allo sprint successivo. Se l'intera processo di feature delivery ha la stessa struttura, un ticket può essere messo in cascata lungo il processo da una fase all'altra seguendo la struttura riportata nella Figura 3. L'unico vincolo sarà quindi la sequenza temporale della rete che non può cambiare in relazione al programma specifico. D'altra parte, ciò non rappresenta un vero ostacolo fintanto che, quando una funzione (o un ticket) viene implementata in NR e viene completata, può essere parcheggiata in attesa che venga inclusa nella prima consegna del nuovo network (NR) disponibile.

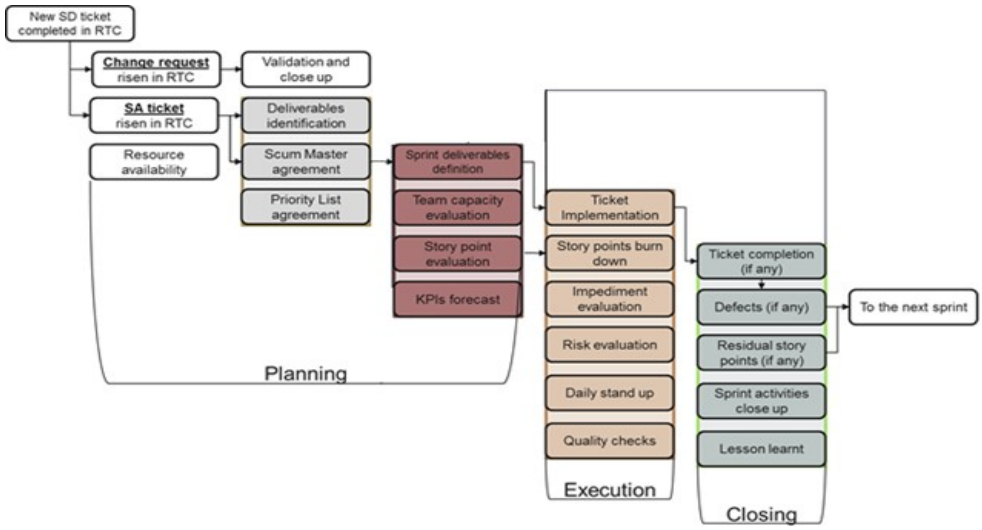

Figura 7 – Processo di feature Delivery con metodologia Agile

TOOLS

Tools di supporto sono utilizzati in diverse fasi del progetto e variano a seconda delle fasi. Durante la fase di definizione, è necessario creare un piano di attività iniziale per avere una previsione del workload. Taliforecastdevono essere utilizzate per definire l'allocazione delle risorse, le stime dei costi o le riunioni delle parti interessate. Quando inizia la fase di implementazione, si deve adottare un monitoraggio più dettagliato al fine di controllare in dettaglio ogni sprint. I dati raccolti saranno utili per la gestione interna del team e delle parti interessate.

PLANNING

Prima di iniziare ogni fase, è necessario mettere in atto una previsione di attività e condurre unallocazione delle risorse in base al volume di attività ed alle capacità delle risorse. In questo modo si ha la possibilità di identificare potenziali problemi relativi allo svolgimento di più task in parallelo oalla mancanza di risorse. Inoltre, le attività di pianificazione consentono di condividere informazioni sul progetto con altre parti rilevanti dell'azienda. Task e milestone sonotraferiti in cascata ad altri team per fornir loro una previsione circa il completamento delle attività, armonizzare i task con quelli di altri stakeholder e informare il management circa lo stato ed i costi di progetto. Alla fine delle attività inerenti ad un determinato processo di delivery, il team crea un archivio storico da utilizzare per le attività future (lessonlernt). Tool di gestione manageriale quali Primavera P6 o MS.Project vengono utilizzati in entrambe le metodologie Waterfall e Agile per prevedere le attività di progetto. L'utilizzo di tool di planning è da vedersi anche come punto di partenza per rivedere stato di avanzamento di progetto, previsione e gestione costi in generale. Le persone e le attività possono essere gestite anche in diversi progetti in base al tempo e ai costi ed alle loro capacità. Pertanto, le risorse potrebbero essere priorizzate in base alle richieste degli stakeholder. La Tabella 4 riassume alcuni degli aspetti positivi e negativi della pianificazione delle attività.

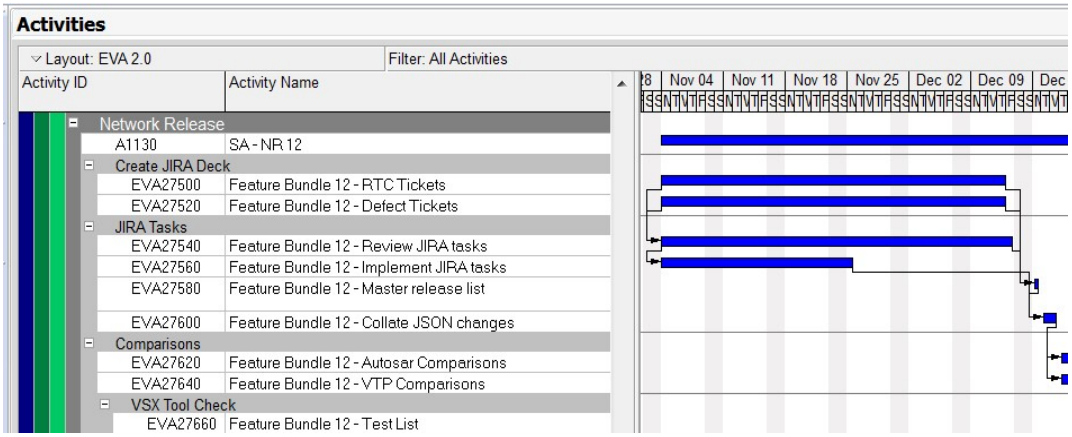

Figura 8 – Esempio di tool per monitoraggio attività

PROs	• Strumento di pianificazione che consente di gestire diversi progetti con diverso livello di complessità • Semplici analisi delle attività (analisi del percorso critico) e facile loro priorizzazione
CONs	• Schematizzazioni delle attività possono sminuire la loro complessità (specialmente per l'approccio Agile) • Sono necessari aggiornamenti costanti per mantenere il piano allineato alle attività

Tabella 4 - PROs and CONsdell'utilizzo di tool di planning

MONITORAGGIO E CONTROLLO

Il non avere uno scopo definito sin dalle prime fasi del progetto implica il doverapplicare un livello più rigoroso di monitoraggio durante tutta la durata delle attività al fine di avere un buon coinvolgimento del team e un regolare controllo degli avanzamenti. Ogni feature viene monitorata, utilizzando strumenti dedicati, in parallelo all'implementazione tecnica per valutare la progressione delle attività ed alleviare i potenziali rischi ed ostacoli che potrebbero crearsi. Il monitoraggio e controllo inizia dalle prime fasi di ogni iterazione con la stima della complessità di ogni ticket e relativa assegnazionedi un tempo di esecuzione adeguato. Ogni attività può essere suddivisa in diversi task e ciascuno di essi può essere implementato da persone diverse (in sequenza o in parallelo) in base alle loro competenze.I tools di monitoraggio e controllo evidenziano le risorse assegnate ad un'attività specifica e il loro stato relativo in differenti momenti della iterazione, le risorse allocate si renderanno automaticamente disponibiliper implementare task successivi quando l'attività viene completata o interrotta. L'uso di questo tool di controllopermette di mantenere un controllo e ridurreeventuali attività non contemplate nello scopo dell'iterazione (scope creep) così come accordato dal cliente e dai vari stakeholder.

Il carico di lavoro totale rispetto alla previsione viene essere mostrato in grafici (burn-down chart) mostranti le attività previste rispetto allo previsione. Utilizzando questo strumento è possibile valutare le attività che sono state finalizzate in un determinato sprint e quindi identificare se stanno procedendo come previsto o se devono essere riprogrammate, Rif. [6]. In maniera analoga si possono utilizzare pie-chart per riportare lo stato complessivo delle attività e i vari stati dei ticket in un preciso momento dell'iterazione. L'uso di dashboard di riassunto è da considerarsi un ulteriore complemento utile per sintetizzare lo stato delle attività e per le comunicazioni con il management.

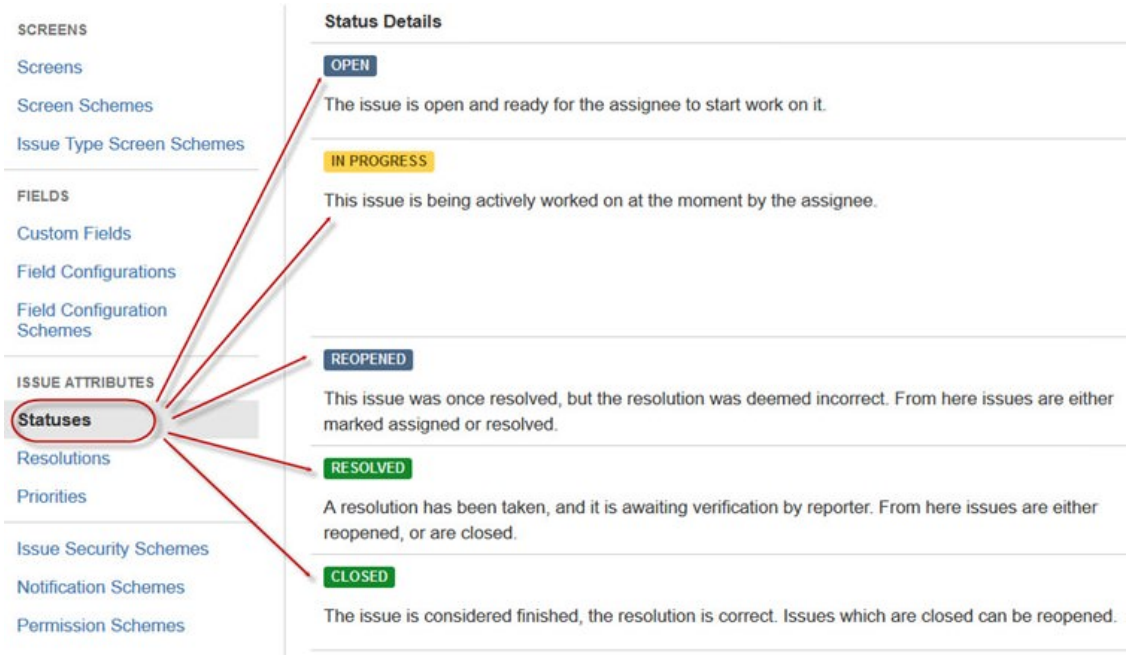

Figura 9 – Monitoring tool for ticket implementation, Rif. [6]

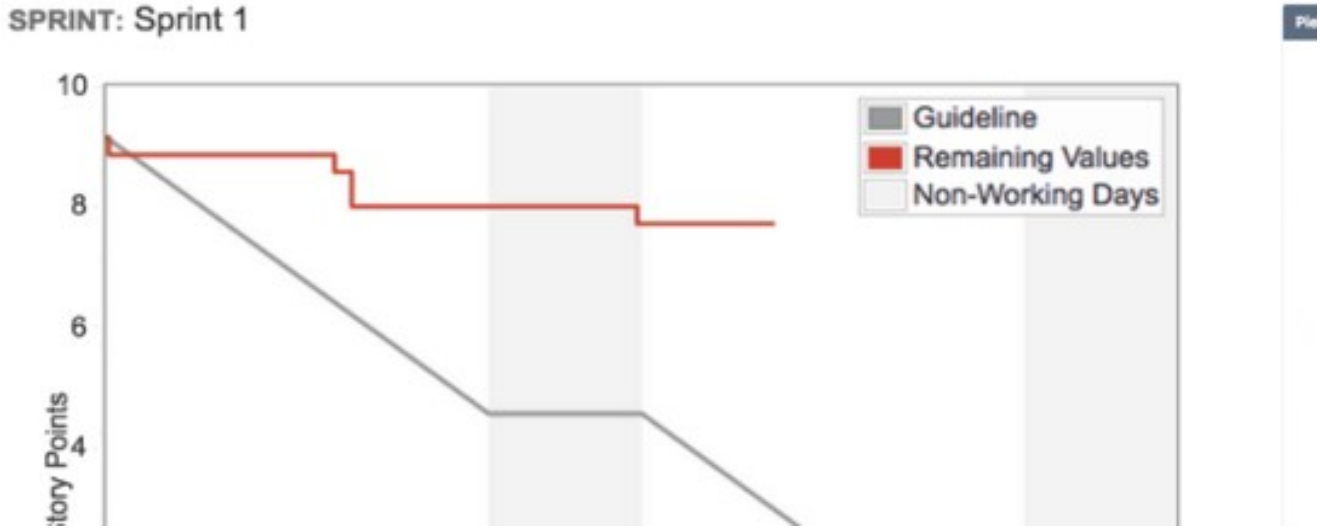

Figura10 – Burndown chart (Forecast vs actual status) e pie chart, Rif.[6]

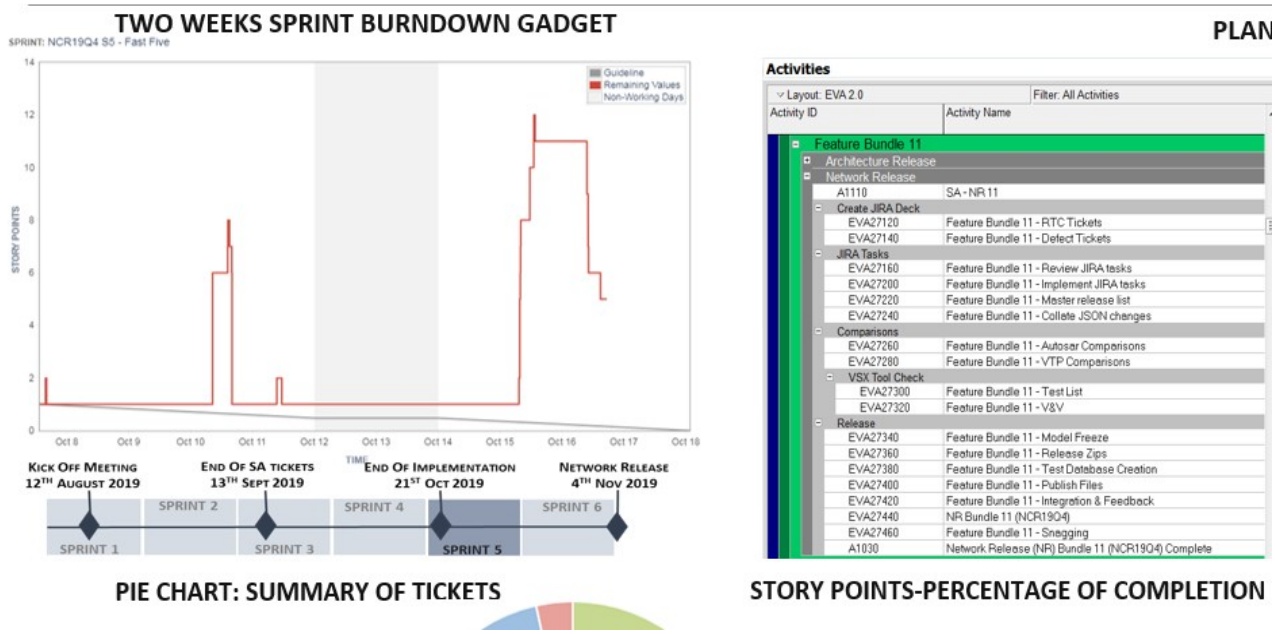

Figura 11 – Dashboard per start-up meeting

CONCLUSIONI

Dal momento che alcuni settori di industrie operanti in settori meccanici, automotive e aeromotive sono stati trasformati in vere e proprie software house che devono supportare il controllo delle parti, tecniche di gestione alternative per soddisfare queste nuove esigenze eper accomodare queste nuove esigenze di mercato devono essere prese in considerazione. L'approccio Agile è sicuramente uno di questi. Tale metodologia è da considerarsi valida quando i requisiti richiedono modifiche durante l'esecuzione del progetto o quando non esiste una definizione completa ad inizio pianificazione. Essere flessibili (o agili) diventa più complesso quando i team iniziano a crescere, quando la definizione non può cambiare o se c'è una mancanza di comunicazione tra le parti interessate. Dal punto di vista della convenienza economica, un approccio tipo Waterfall richiede costi inferiori in quei casi in cui la definizione può essere completata (o quasi) nelle fasi iniziali. In sistemi complessi si può considerare l'uso di una metodologia combinata, correttamente armonizzata, che porti ad ultimare un programma complesso richiedente l'implementazione di progetti aventi diversa natura ma uno stesso fine. Come esempio si consideri la realizzazione di un veicolo in cui elettronica, architettura di sistema e parti meccaniche devono venire armonizzate per assicurare un design di appeal ma anche una affidabilità degna di tale design e delle feature di controllo che esso contiene.

REFERENZE

1. Murray Robinson, Agile Technology Delivery Process, March 23, 2010
2. www.startnearshoring.com/knowledge/it-project-management-a-quick-guide-to-tools-and-methodologies/
3. F. Turkley, Prince 2 Foundation training manual, 2017
4. hub.packtpub.com/9-reasons-to-choose-agile-methodology-for-mobile-app-development
5. T. Linchpin, A Beginner's Guide To The Agile Method & Scrums 2019
6. www.atlassian.com/agile/tutorials
7. J. Highsmith, Agile project management: creating innovating products 2nd Ed., 10 Jul 2009
8. www.visual-paradigm.com/scrum/classical-vs-agile-project-management

9. M. Cohn, Agile estimating and planning, 1 Nov 2005