ALMA MATER STUDIORUM UNIVERSITA' DI BOLOGNA

CAMPUS DI BOLOGNA - SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il Management (classe L-31)

Tesi di Laurea

in Ingegneria del Software

APPLICAZIONE DELLA METODOLOGIA SCRUM IN AMBIENTI DISTRIBUITI E SU LARGA SCALA

Relatore: **Laureando:**

Prof. Davide Rossi **Giovanni Enrico Ceglia**

Anno Accademico 2020/2021

Sessione straordinaria

Contesto

Le metodologie agili sono diventate l'approccio standard allo sviluppo del software. Ventuno anni dopo la pubblicazione del Manifesto dello sviluppo del software agile è emerso che il metodo più popolare ed utilizzato è Scrum, un framework flessibile che risponde all'imprevedibilità in modo davvero efficace. In origine il suo design fu pensato per gruppi composti da pochi sviluppatori co-localizzati; successivamente è stato adottato in contesti più ampi, distribuiti o diversi dal dominio dell'ingegneria del software per la sua semplicità ed efficacia, provocando una necessità di adattamenti alle varie circostanze. La personalizzazione del metodo ha generato la nascita di nuovi artefatti, nuovi ruoli e nuovi eventi oltre che ad una modifica della molteplicità dei diversi aspetti del metodo in base alle esigenze. In questa tesi si vuole trattare l'adattamento di Scrum in contesti di sviluppo su larga scala o distribuito.

Introduzione

Pianificare, conoscere e prevedere sono aspetti chiave per la riduzione del rischio di fallimento di un progetto o di un'intera Azienda. Per non incorrere in tali rischi le imprese devono affidarsi al project management. In ambito di ingegneria del software sono le metodologie agili ad eccellere negli ultimi anni; tuttavia, sono state progettate esclusivamente per gruppi composti da un ristretto numero di sviluppatori co-localizzati. Negli ultimi anni lo sviluppo software distribuito è diventato una realtà aziendale comune. La principale causa della separazione fisica dei team è l'aumento delle dimensioni del team di sviluppo che mette a dura prova il coordinamento; pertanto, una strategia comune è quella di dividere gli sviluppatori in diversi team organizzati in strutture come "Scrum of Scrum". Altre cause sono "l'offshoring e l'outsourcing" da parte delle aziende che adottano queste pratiche per migliorare le prestazioni. Questo cambio di scenario genera contrasti con molti dei presupposti chiave all'interno dello sviluppo agile che richiede quindi degli adattamenti. La tesi intende analizzare il framework Scrum, le modifiche al metodo originale e le cause che hanno generato questa esigenza di adattamenti.

Indice

CAPITOLO 1: Descrizione di Scrum

1.1 Cenni Storici

Scrum è un framework utilizzato per gestire lo sviluppo di processi complessi. Affonda le radici delle sue origini in Giappone: il termine “Scrum” è stato introdotto da Tekeuchi e Nonaka che nel 1986 descrissero un nuovo approccio allo sviluppo di prodotti commerciali che avrebbe aumentato la velocità e la flessibilità. Il termine Scrum descrive la fase di mischia nel gioco del rugby, usata per riprendere il gioco dopo un'infrazione. Questa analogia deriva dal lavoro di squadra adottato dal team di sviluppo che adotta questa metodologia, cercando di raggiungere l'obiettivo come unità. Fu descritto nel 1995 da Jeff Sutherland e Ken Schwaber in un articolo che lo presentava come una metodologia che incarna principi di sviluppo iterativo e incrementale. Negli anni successivi fu integrato con alcune best-practice dell'industria proprio da Sutherland e Schwaber sfruttando le loro esperienze lavorative. Nel febbraio del 2001 diciassette esperti dello sviluppo software si incontrarono per cercare delle metodologie alternative allo sviluppo tradizionale. Tutti i partecipanti erano concordi nel ritenere che le metodologie derivate dal modello a cascata non rappresentassero al meglio la soluzione più efficace in ambienti moderni, dove era importante possedere una certa agilità nell'affrontare la costruzione di sistemi software. Il cliente doveva essere reso partecipe per capire meglio i requisiti della soluzione desiderata e per instaurare un forte legame di collaborazione con il team di sviluppo. In questo modo la soluzione finale risulta essere

compatibile alle esigenze del cliente, anche in caso di eventuali modifiche alle richieste iniziali. Quindi l'agilità accennata precedentemente consiste nella capacità da parte del team di sviluppo di prevedere e sfruttare al meglio i cambiamenti dei requisiti adattandosi velocemente. Essere agili significa inoltre alleggerire i processi: le documentazioni approfondite del sistema descrivono al meglio ogni aspetto e permettono una migliore familiarizzazione al progetto per ogni membro, ma allo stesso tempo possono rappresentare un ostacolo. Un componente del team che ha bisogno di essere istruito o aggiornato sul progetto in corso, potrebbe necessitare di troppo tempo per essere informato. In situazioni turbolente, dove anche pochi giorni di ritardo fanno la differenza, la velocità di comprensione assume un ruolo chiave. Il risultato dell'incontro fu la redazione del Manifesto Agile, un insieme di valori e principi comuni per il coordinamento di ambienti in ambito di sviluppo software, in cui si predilige la collaborazione con il cliente, le relazioni e le comunicazioni fra gli attori del progetto, la risposta al cambiamento, la priorità al funzionamento del software stesso piuttosto che la realizzazione di una documentazione esaustiva. I firmatari del Manifesto Agile avevano lo scopo di diffondere le idee per nuovi e più realistici approcci alla realizzazione di prodotti software.

1.2 Fondamenti teorici

Come tutte le altre metodologie agili, Scrum prevede eventi, artefatti e ruoli. Ogni singolo aspetto serve ad uno specifico scopo ed è essenziale per il successo del suo utilizzo. Si basa sulla teoria empirica del controllo dei processi privilegiando la trasparenza, l'ispezione e l'adattamento. Scrum è una metodologia trasparente perché il processo ed il risultato di un lavoro risulta visibile a tutti i partecipanti. La trasparenza richiede che gli aspetti significativi del processo siano definiti da uno standard comune per far si che gli osservatori possano condividere una comune comprensione di ciò che viene valutato. Gli osservatori vengono denominati Scrum Master e si occupano di ispezionare le attività dei partecipanti misurando l'avanzamento verso un obiettivo con lo scopo di rilevare le eventuali deviazioni indesiderate. Queste ispezioni però non devono essere frequenti per non intralciare il lavoro stesso e risultano quindi più utili quando sono eseguite diligentemente da chi ha l'abilità e la competenza necessaria per effettuarle in un particolare stadio del lavoro. Nel caso in cui gli Scrum Master evidenzino un aspetto al di fuori dei limiti accettabili che possa rendere il risultato finale compromesso, il metodo deve adattarsi rapidamente al fine di ridurre al minimo questo rischio. La partecipazione attiva da parte del cliente gli permette di influire sul lavoro. L'approccio non tenta di definire tutto il coordinamento in principio, al contrario permette al team di lavorare in brevi iterazioni perfezionando il piano nel corso del

tempo. Le regole di Scrum legano insieme gli eventi, gli artefatti ed i ruoli governando le relazioni e le interazioni tra essi.

1.3 Ruoli

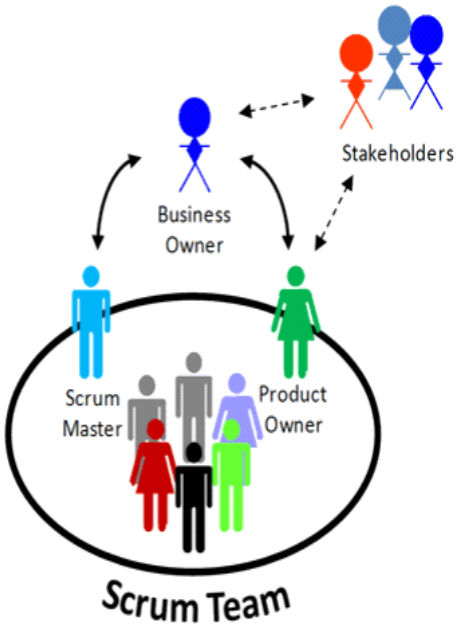

Fig.1 – Componenti di uno Scrum Team.

I ruoli che compongono il Team Scrum che si occupa della realizzazione del prodotto sono: il Product Owner, il Development Team ed uno Scrum Master. Il Product Owner rappresenta la voce del cliente ed ha il compito di massimizzare il valore del prodotto ed il lavoro svolto dal Development Team. Ha la responsabilità esclusiva di gestione del Product Backlog, un documento costituito da un elenco ordinato di idee per il prodotto. Deve far si che gli elementi che compongono il Product backlog siano espressi in

modo chiaro e che siano ordinati per raggiungere nel breve gli obiettivi prefissati. Funge quindi da intermediario fra cliente e team di sviluppo, ruolo costituito da un'unica persona, la quale ha la facoltà di scegliere se assecondare o meno le richieste dei componenti del team con il fine di ottimizzare il valore del prodotto e ridurre i tempi di esecuzione. Lo Scrum Master è l'intermediario fra il Product Owner ed il team di sviluppo, il suo compito, infatti, è quello di collaborare con entrambe le parti. Ha la responsabilità di aiutare il Product Owner nella gestione del Product Backlog e di limitare gli ostacoli che riducono la capacità del team di raggiungere il risultato. Il Development Team è costituito da un insieme di professionisti che lavorano per produrre un incremento di prodotto potenzialmente rilasciabile alla fine di ogni Sprint. La sua dimensione ottimale è abbastanza piccola da rimanere agile e abbastanza grande da completare il lavoro nel minor tempo possibile, di solito è composta da un numero variabile di elementi, che vanno dai tre ai nove circa, e non comprende i ruoli del Product Owner e dello Scrum Master. Avere meno professionisti comporterebbe un minore guadagno in termini di produttività; mentre avere troppi attori richiederebbe un eccessivo lavoro di coordinamento. Gli elementi del Team sono auto-organizzati, si occupano di trasformare il Product Backlog in incrementi di prodotto autonomamente.

1.4 Eventi

Gli eventi sono occasioni per pianificare ed ispezionare il lavoro del team. Scrum utilizza eventi time-box per assicurarsi una quantità di tempo appropriata ed evitare sprechi nel processo di pianificazione. L'unità di base è lo Sprint, di durata fissa che generalmente varia da una a quattro settimane. Ogni Sprint è preceduto da una riunione in cui vengono identificati gli obiettivi denominati Sprint Goal e vengono stimati i tempi. Servono a limitare il rischio ad un mese di costo ed assicurare che l'ispezione sia eseguita almeno una volta al mese. Può essere considerato come un contenitore di altri eventi, infatti è costituito dallo sprint Planning, Daily Scrum, lavoro di sviluppo, Sprint Review e dallo Sprint Retrospective. Lo Sprint Planning è un incontro fra Product Owner, Scrum Master e Development Team nel quale si discute sul lavoro che deve essere svolto durante lo Sprint che si sta pianificando. Dopo aver individuato lo Sprint Goal, costituito da elementi selezionati dal Product Backlog, il team di sviluppo decide come lavorare a queste funzionalità per costituire un incremento del lavoro svolto. Alla fine di questa riunione il Product Owner e lo scrum Master devono essere informati su come il team intenda lavorare. Il Daily Scrum è una riunione con una durata massima di quindici minuti, tenuta all'inizio di ogni giornata durante lo Sprint. In questo meeting quotidiano i membri del Team descrivono il lavoro svolto il giorno precedente, il lavoro che verrà svolto il giorno stesso ed eventuali ostacoli riscontrati. I partecipanti solitamente sono tenuti a rimanere in piedi al fine

di evitare distrazioni. Alla fine di ogni Sprint si tiene lo Sprint Review, un incontro della durata massima di quattro ore che serve ad ispezionare l'incremento e adattare il Product backlog. Fra una Sprint Review e la successiva Sprint Planning si tiene la riunione per la Sprint Retrospective, util ai componenti dell'intero Scrum Team pe auto-ispezionarsi e per creare un piano di miglioramento da attuare durante il prossimo Sprint.

1.5 Artefatti

Gli artefatti di Scrum rappresentano il lavoro al fine di fornire trasparenza, opportunità di ispezione ed adattamento, ma hanno anche lo scopo di fungere da strumento per facilitare il raggiungimento di un obiettivo. Il fattore comune di tutti gli artefatti è la trasparenza delle informazioni chiave. Il Product Backlog è un elenco ordinato di requisiti per le modifiche da apportare al prodotto. I vari elementi che compongono questo elenco non smettono mai di cambiare e rendono quindi l'artefatto dinamico; sono caratterizzati da: descrizione, stima e valore. Ogni Sprint ha il suo backlog, che definisce l'insieme delle operazioni da svolgere all'interno del singolo Sprint. Queste operazioni rappresentano un incremento del prodotto e devono avere una qualità sufficientemente elevata da consentirne il rilascio al cliente.

CAPITOLO 2: Esigenza di adattamento

Nel corso degli anni il framework Scrum è diventato il più popolare fra le metodologie agili per la sua semplicità ed efficacia. È decollato in particolar modo nell'ambito di sviluppo software, ma è stato adottato anche da altre discipline al di fuori del dominio per cui è stato pensato. Dalla pubblicazione del Manifesto agile ad oggi sono cambiate molte cose. Dal 2020 a questa parte le aziende hanno dovuto affrontare una pandemia che ha messo a dura prova gli equilibri che garantiscono solidità, sono state infatti costrette a modificare le proprie pratiche. Le grandi aziende di successo che dedicano il loro impegno nello sviluppo di software hanno dovuto, inoltre, adattare le proprie dimensioni a causa dell'aumento della mole di lavoro. La popolarità di Scrum nelle varie aziende permette di raccogliere dati empirici sul suo utilizzo, per consolidare le conoscenze pratiche ma anche per determinare quali parti della metodologia pongono problemi nell'industria. Riscontrare problematiche è naturale quando si applica una metodologia di coordinamento pensata per un dominio diverso dal proprio e, di conseguenza, le aziende tendono a rivisitare alcuni degli aspetti di Scrum per evitare queste complicazioni e migliorare le prestazioni in termini di risultati e di costi. L'articolo di Michal Horn e di Nikolaus Obwegeser (2022) individua le principali cause della necessità di adattamento ed alcune strategie utilizzate per supportare le pratiche previste dalla metodologia Scrum in cui uno studio basato su una revisione su larga scala della

letteratura esistente, in cui sono emerse nove obiettivi e sette strategie di modifica. Lo scopo dell'articolo è effettuare quindi una revisione che consenta l'identificazione, la sistematizzazione e la valutazione delle pratiche canoniche della metodologia Scrum. Gli autori hanno deciso di adottare un approccio sistematico per garantire trasparenza tramite un'approfondita fase di analisi e filtraggio. Il campione di letteratura è stato raccolto utilizzando una query strutturata su Scrum ai database Scopus, Web of Science e IEEE Explore. La query eseguita nell'agosto del 2020 ha restituito 3145 documenti che hanno costituito la base per il processo di filtraggio. Dopo l'eliminazione dei documenti duplicati, il numero degli articoli che componevano il campione iniziale è sceso a 2209 articoli i quali sono stati sottoposti alle procedure di filtraggio. Questa fase è rappresentata da due proiezioni: proiezione del titolo e proiezione astratta. Per quanto riguarda la prima, gli autori hanno esaminato separatamente i titoli degli stessi cento articoli per rafforzare l'affidabilità del processo di codifica. Il grado di affidabilità è stato misurato utilizzando il Kappa di Cohen, con il risultato di 0.96 che indica un elevato grado di accordo. Il coefficiente Kappa è stato calcolato come segue:

Coefficiente Kappa

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} = \frac{0.98 - 0.50}{1 - 0.50} = 0.96$$

Pr(a) rappresenta l'accordo relativo fra gli autori mentre Pr(e) è l'ipotetica probabilità di accordo casuale. Successivamente è stato effettuato il filtraggio tramite lo stesso criterio dei restanti articoli. Il risultato è che dei 2209 articoli, 1423 sono risultati potenzialmente rilevanti sulla base dei soli titoli. La proiezione astratta è costituita da due condizioni: il documento deve introdurre una modifica e deve fornire una spiegazione sulla causa di necessità di modifica. Gli autori, in questo modo, hanno potuto identificare i nove obiettivi di modifica generici e le sette strategie di modifica comuni.

Obiettivi:

1)Prestazione

L'obiettivo più frequente per la modifica di Scrum è raggiungere prestazioni elevate, quindi, migliorare la qualità del prodotto e ridurre i costi.

2) Contesto

Il secondo obiettivo più comune è quello di accogliere un contesto specifico. Scrum offre un approccio al lavoro particolarmente semplice e può essere trasposto in una serie di impostazioni diverse.

3) Giustapposizione

Obiettivo di combinare la metodologia Scrum ad un altro framework. Questa esigenza può essere dovuta a esigenze istituzionali, motivate dalla volontà di trarre vantaggio da entrambe le metodologie.

4) Architettura

Questa categoria aggrega il desiderio comune di specificare l'esito del processo di sviluppo, almeno ad un livello elevato. Sebbene Scrum originariamente suggerisca che la pianificazione di alto livello e l'architettura del software siano condotte in uno sprint che precede lo sviluppo effettivo, molti professionisti vedono la necessità di occuparsi di quei processi con maggiore attenzione.

5) Sviluppo Distribuito

Questa categoria descrive tutti gli sforzi volti all'utilizzo di Scrum in contesti non collocati ma geograficamente distribuiti.

6) Estensioni Manageriali

L'obiettivo "Estensioni manageriali" mira a combinare l'aggiunta di livelli più elevati di gestione del prodotto software con l'integrazione di strumenti gestionali per il coordinamento con l'obiettivo di dotare il processo di sviluppo di maggiore responsabilità e controllo.

7) Esperienza Utente

I metodi di sviluppo Agile, e Scrum, in particolare, mirano a raggiungere elevati livelli di usabilità coinvolgendo l'utente nel processo di sviluppo, oltre a specificare l'accettazione da parte dell'utente come condizione finale, che deve essere superata affinché un elemento del backlog sia considerato

completato. Tuttavia, l’obiettivo di questa categoria è quello di coinvolgere l’utente anche in ambito di progettazione.

8) Ridimensionamento

Oggi, Scrum viene spesso applicato ad una mole di lavoro che supera le capacità di un singolo team Scrum. Nei progetti industriali per lo sviluppo di grandi artefatti software, molti team di solito contribuiscono allo sviluppo del prodotto complessivo. L'aumento delle dimensioni del team di sviluppo mette a dura prova il coordinamento.

9) Sicurezza

Scrum non si occupa esplicitamente di sicurezza ma tuttavia ci sono numerosi software che richiedono un livello di sicurezza elevato. L’obiettivo è quello di affrontare la sicurezza in Scrum tramite metodi sistematici.

Strategie di modifica:

1) Guida al metodo

   Appelli ai principi del metodo selezionato per affrontare al meglio determinate pratiche ed evitare quelle ritenute dannose.

2) Procedure

   Strategia che si basa sull’introduzione di nuove attività nel metodo che possono anche ripresentarsi più volte.

3) Infusione

   Descrive una riconsiderazione approfondita di molteplici aspetti di Scrum, basandosi su altri processi.

4) Strumenti

   Strategia che non intende modificare direttamente la metodologia originale ma vuole supportare tramite l'utilizzo di nuovi strumenti.

5) Artefatti

   Categoria che include l'utilizzo di nuovi artefatti di supporto agli artefatti previsti da Scrum.

6) Pre-sviluppo

   Si riferisce all'introduzione di processi, artefatti e talvolta ruoli aggiuntivi che si occupano specificatamente di attività che anticipano l'inizio dello sviluppo.

7) Molteplicità

   Categoria di strategie che sfruttano la molteplicità di un artefatto, di un ruolo o di un intero team.

La personalizzazione dei metodi sono fenomeni comuni e ben riconosciuti all'interno del dominio dell'ingegneria del software poiché la complessità e le contingenze dei diversi contesti del mondo reale possono difficilmente riflettersi in un approccio metodologico unificato. Tuttavia, alcuni evangelisti di Scrum sostengono fortemente di seguire le pratiche delineate "da

manuale" per evitare di introdurre le stesse inefficienze ed elementi disfunzionali che ci si proponeva di eliminare introducendolo. Un'altra causa che genera questa suggestione a rivisitare il metodo, oltre alla sua applicazione in contesti diversi da quello per cui è stato pensato, è l'espansione che un'industria può subire nel corso del tempo. Con l'aumentare della mole di lavoro che un'impresa deve affrontare, è indispensabile aumentare anche il numero di componenti e, talvolta, risulta necessario utilizzare più squadre in modo tale da poter svolgere diversi compiti separatamente e ridurre i tempi di rilascio. Tuttavia, l'offshoring e l'outsourcing sono le ragioni più comuni per cui molti team di sviluppo oggi sono distribuiti. Sono due strategie distinte che mirano a migliorare le prestazioni di un'azienda, ma al contempo generano il dislocamento dei team che potrebbe provocare difficoltà in fase di coordinamento. Quando un'impresa non è co-localizzata ed i vari team sono distribuiti in diverse aree geografiche, le difficoltà nel coordinamento possono derivare dalla pianificazione tra fusi orari diversi, dalle barriere linguistiche e da problemi di comunicazione interculturale, oltre che dalla ridotta larghezza di banda di comunicazione. I metodi agili, come Scrum, sono stati originariamente progettati per essere utilizzati da singoli piccoli team, i cui membri sono collocati, lavorando faccia a faccia, preferibilmente nelle stanze del team. Al giorno d'oggi, molte aziende che sviluppano sistemi di grandi dimensioni con più team distribuiti in diverse località geografiche vorrebbero sfruttare i vantaggi dei metodi agili. Pertanto, è necessario ridimensionare le pratiche

agili. Il ridimensionamento comporta diverse sfide, come il coordinamento tra i team, la mancanza di architettura, la mancanza di analisi dei requisiti e tutte le sfide dei progetti distribuiti. Numerosi studi discutono di come nuove procedure o particolari forme di molteplicità possano supportare Scrum in un ambiente distribuito. Nel complesso, la maggior parte degli studi raccomanda la moltiplicazione dei team e talvolta delle procedure, mentre si consiglia generalmente di condividere i principali artefatti di Scrum tra i team. I metodi agili sono costruiti attorno a team responsabilizzati e auto-organizzati con una forte attenzione alla collaborazione e alla comunicazione supportate da varie pratiche agili, tra cui confronto, collaborazione con i clienti, stand-up, recensioni, retrospettive e gioco di pianificazione. Pertanto, il loro utilizzo in ambienti distribuiti richiede delle modifiche.

CAPITOLO 3: Adattamenti

Lo sviluppo software globale promette risparmi sui costi, accesso a una grande forza lavoro multi-qualificata, delle tempistiche ridotte e la possibilità di seguire le attività di sviluppo software 24 ore su 24. Questi fattori decisivi, tra gli altri, hanno reso lo sviluppo software globale una realtà quotidiana nelle organizzazioni di oggi, sebbene gli ambienti di sviluppo siano più complessi e presentino diverse sfide come: distanze fisiche e fusi orari, perdita di "squadra", differenze culturali, questioni strategiche, differenze di processo, gestione della conoscenza e sfide tecniche. Il project management, in questo tipo di imprese, è il ruolo più delicato da rivestire non solo per la varietà di aree geografiche occupate, ma soprattutto per l'elevato numero di dipendenti da gestire. I metodi agili sono stati originariamente progettati per piccoli team co-localizzati. A causa della loro popolarità, oggigiorno vengono applicati anche da grandi aziende in grandi progetti di sviluppo software che impiegano più team distribuiti in diverse località geografiche. Il ridimensionamento dei metodi agili in questo nuovo contesto introduce nuove sfide, come il coordinamento tra i team, la distribuzione del lavoro senza un'architettura definita o requisiti adeguatamente definiti, nonché tutte le sfide dei progetti distribuiti. Le aziende che intendono sfruttare i vantaggi che la metodologia Scrum ha da offrire hanno bisogno di rivisitare i fondamenti teorici. Pertanto, nel corso degli anni, il panorama del project management attraverso la metodologia

Scrum ha visto espandere il proprio insieme di pratiche con nuovi artefatti, nuovi eventi e soprattutto nuovi ruoli.

2.1 Nuovi strumenti

- **DBoard**

Uno degli strumenti più interessanti utilizzati in ambito di project management con lo scopo di supportare le pratiche della metodologia Scrum applicata a contesti di sviluppo software globale è la dBoard. La dBoard funge da bacheca digitale per Scrum; descritta nell'articolo scientifico di Esbensen et al. (2015), è progettata come una "finestra virtuale" tra aree geografiche diverse occupate dal team Scrum. Collega due location con video e audio live, a cui è sovrapposta una lavagna Scrum digitale sincronizzata e interattiva che adatta la fedeltà del video/audio alla presenza delle persone che si trovano davanti. La dBoard è progettata per funzionare come divulgatore di informazioni da cui è facile ottenere una panoramica dello stato del lavoro, come uno spazio multimediale che fornisce consapevolezza sulla presenza di colleghi di lavoro a distanza e come strumento di supporto attivo per le riunioni.

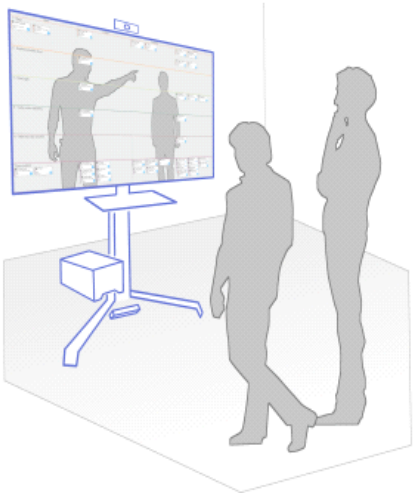

Fig.2 – La dBoard è uno strumento di supporto per videoconferenze progettato per team distribuiti.

Questo strumento non solo permette di migliorare la qualità di una videoconferenza, ma presenta anche una serie di funzionalità che forniscono un ulteriore supporto al coordinamento di diversi team dislocati. La Scrum Board è una funzionalità che permette di tenere traccia delle attività del product backlog, supportando la manipolazione diretta come una vera e propria bacheca, organizzata in task che sono rappresentati come piccoli post-it digitali disposti in colonne che ne indicano lo stato e righe che rappresentano la user story a cui appartengono. Il suo hardware è costituito da un ampio display da 65” con una risoluzione di 3840 x 2160, da una fotocamera ad alta definizione collegata e un microfono di qualità cattura video e audio. Tutto l'hardware è montato su un supporto mobile e tutti i componenti sono collegati a un computer senza ventola fissato al supporto. La configurazione garantisce che l'installazione e l'avvio di dBoard sia semplice come spostarla nella posizione desiderata e collegare

l'alimentazione. Il sensore Kinect v2 utilizzato per rilevare la vicinanza alla scheda è montato tra le ruote del supporto. In alternativa, lo schermo può essere appeso a una parete. Il frontend dello strumento è un'applicazione HTML5 e JavaScript creata con AngularJS e progettata per Google Chrome. Due dBoard collegate inviano e ricevono video e audio utilizzando WebRTC, che è un framework JavaScript basato sul peer-to-peer mentre tutte le altre informazioni come attività, storie degli utenti ed informazioni sullo stato, vengono comunicate al server tramite websocket.

- **Knowledge4Scrum**

Un altro strumento da citare è la Knowledge4Scrum, descritta nell'articolo di Sungkur e Ramasawmy (2014), è uno strumento di gestione della conoscenza per team agili distribuiti. L'introduzione dello sviluppo software globale comporta una serie di difficoltà, soprattutto legate alla condivisione delle conoscenze. Ad esempio, all'interno di tali team si osserva spesso mancanza di trasparenza, per cui un membro del team è totalmente all'oscuro delle attività dei suoi colleghi. In secondo luogo, l'indisponibilità dei membri del team a causa delle differenze di fuso orario si aggiunge all'elenco dei problemi affrontati dai team distribuiti. Terzo, ci possono essere incomprensioni tra i membri del team a causa di problemi di comunicazione, specialmente nei casi in cui la lingua madre dei membri del team è diversa. Ulteriore problema comune, affrontato dai team distribuiti,

è la perdita di conoscenza quando un dipendente si dimette dal suo incarico. Per sopperire ai problemi sopra descritti, questo strumento si propone come supporto per la condivisione delle conoscenze. Una delle principali distinzioni fatte nella gestione della conoscenza è tra conoscenza esplicita e conoscenza tacita: la condivisione tacita della conoscenza avviene normalmente durante le interazioni tra le persone, mentre la conoscenza esplicita riguarda la conoscenza trasmissibile tramite linguaggio formale. La conoscenza tacita di proprietà di un “knowledge worker” può essere resa esplicita, ma la creazione di conoscenza esplicita, come ad esempio tramite documentazione, produce dei costi. D'altra parte, la mancanza di una conoscenza esplicita ostacola la comprensione e produce anche dei costi. I metodi agili si basano fortemente sulla conoscenza tacita. La capacità degli agilisti di gestire le proprie conoscenze è, quindi, una preoccupazione chiave nei metodi agili. Con il processo globale di sviluppo del software che diventa sempre più complesso, gli sviluppatori di software non possono fare affidamento solo sulla loro tacita conoscenza per risolvere completamente i problemi, senza documentazione e supporto esplicito della conoscenza, il gruppo agile potrebbe non funzionare in modo efficiente. Di conseguenza, i professionisti del metodo agile hanno un urgente bisogno di supporto per la gestione della conoscenza nelle loro organizzazioni. Anche Knowledge4Scrum consiste in una Scrum Board virtuale che funge da repository globale. Le sue funzionalità sono quelle di fornire ai membri dei vari team le attività quotidiane da svolgere. Inoltre, permette la ricerca e il

reperimento di informazioni attraverso allegati e forum di discussione. Un altro aspetto principale dello strumento risiede nella sezione relativa alla creazione e all'applicazione della conoscenza, in cui sono state utilizzate numerose tecniche di data mining per identificare tendenze e modelli nei dati raccolti in passato. Il data mining è un processo con l'obiettivo di ordinare grandi quantità di dati e scoprire nuove informazioni o conoscenze.

2.2 Nuovi eventi

A causa della separazione fisica dei team di sviluppo in software globale, molti dei presupposti chiave all'interno dello sviluppo agile come l'interazione con i clienti, la comunicazione del team e l'essere faccia a faccia, non reggono. Per ottenere il vantaggio dallo sviluppo agile, le pratiche devono essere modificate quando applicate alle impostazioni distribuite. Potrebbero inoltre essere necessarie pratiche di supporto a quelle che caratterizzano la metodologia originale. Le pratiche distribuite di supporto si sforzano di fornire soluzioni alle sfide che la distanza geografica pone allo Scrum Team. Queste operazioni sono descritte nell'articolo di Paasivara et al. (2009) che offrono un elenco di attività concrete di supporto per facilitare l'applicazione di Scrum in ambienti distribuiti, come la riduzione dell'indipendenza dei siti e la sincronizzazione dell'orario di lavoro. Una delle pratiche descritte nell'articolo è la visita frequente, utilizzate per creare e mantenere la fiducia e migliorare la collaborazione. Le visite

effettuate all'inizio del progetto mirano a costruire una relazione, mentre quelle svolte durante lo svolgimento del lavoro sono più brevi e mirano a mantenere la collaborazione. I membri del team dovrebbero ruotare continuamente tra i siti e almeno un membro del team alla volta dovrebbe essere in visita. Spesso le imprese tendono ad inviare i propri ingegneri esperti da un sito all'altro per un periodo di tempo più lungo; in questo modo possono riferire le lezioni apprese e stabilire le direzioni future per i progetti oltre che tendere a fornire formazione iniziale e tutoraggio all'altro sito. Un'altra pratica di supporto è quella di realizzare squadre simmetriche dove ogni elemento di una squadra ha la sua controparte in un'altra. Questa pratica riduce la dipendenza fra i vari siti. Inoltre, le industrie che vogliono adottare la metodologia Scrum in ambienti distribuiti dovrebbero fornire diversi tipi di comunicazione applicabili anche in parallelo, ad esempio servizi di messaggistica istantanea, Wiki e condivisione degli schermi. Un altro vantaggio potrebbe scaturire dalla massimizzazione delle ore di lavoro sovrapposte per garantire una comunicazione costante.

2.3 Nuovi ruoli

L'aumento delle dimensioni del team di sviluppo mette a dura prova il project management. Pertanto, una strategia comune è quella di dividere gli sviluppatori in diversi siti moltiplicando il numero dei ruoli previsti dall'originale definizione di Scrum Team. Questi team vengono quindi organizzati in strutture come ''Scrum of Scrums'' che richiede un'infrastruttura di supporto aggiuntiva sotto forma di nuove procedure e ruoli che hanno il compito di garantire il coordinamento fra i vari siti. Nell'articolo di Paasivara et al. (2012) viene descritta l'Area Product Owners per ridimensionare il ruolo di product owner. Ciascun Area Product Owners è responsabile delle funzionalità in un'area di prodotto specifica e lavora con i team che hanno sviluppato tali funzionalità. L'Area è composta da due persone: un architetto di sistema e un rappresentante della gestione del prodotto, due ruoli ben distinti ma in costante collaborazione. L'architetto del sistema lavora a stretto contatto con i team, al contrario del rappresentante della gestione del prodotto. Nella sede principale, l'architetto del sistema si trovava allo stesso piano dei team, mentre il rappresentante della gestione del prodotto si trova in un altro edificio. Le Area Product Owners descritte nell'articolo non sostituiscono la figura del Product Owner dell'intero progetto ma lo supportano specializzandosi nelle proprie competenze.

CAPITOLO 4: Casi Studio

Di seguito vengono illustrati alcuni Casi Studio su alcune delle novità introdotte dalle imprese per migliorare il coordinamento in ambienti distribuiti o su larga scala.

- **DBoard**

L'articolo di Esbensen et al. (2015) fornisce una descrizione dettagliata della dBoard ma ci permette anche di analizzare un caso studio sul suo utilizzo per misurarne l'utilità. Lo studio è stato condotto invitando delle società di sviluppo software che utilizzano Scrum a partecipare ad un workshop di valutazione. Allo studio hanno partecipato sette ingegneri informatici di tre diverse aziende con un'età media pari a 30,5. I partecipanti hanno riferito di essere praticanti delle metodologie Agile e Scrum ad un livello esperto, la loro anzianità varia da 6 mesi a 10 anni. In particolare, due dei partecipanti erano Scrum Master, due Product Owner e il resto erano sviluppatori o tester.

Il workshop è stato suddiviso in quattro parti:

i. un'introduzione alla dBoard in cui i partecipanti sono stati accolti allo studio, firmato un modulo di consenso informato, compilato un questionario demografico e hanno ricevuto una presentazione dettagliata delle caratteristiche di dBoard;

ii. una sessione pratica in cui i partecipanti sono stati invitati a sperimentare le due diverse dBoard;
iii. una sessione in cui è stato chiesto ai partecipanti di agire in due diversi scenari;
iv. una discussione di gruppo conclusiva in cui ai partecipanti, dopo aver completato un breve sondaggio, è stato chiesto di riflettere sul dBoard ed elaborare le loro risposte al sondaggio.

I due scenari sono stati scelti tra le attività più comuni nei tradizionali team Scrum distribuiti.

Nel primo scenario è stato richiesto ai partecipanti di eseguire prima un semplice compito di progettazione (ad esempio, selezionare un modello per la progettazione di un sito Web basato su ispirazioni da siti Web esistenti); poi di aggiornare lo stato dell'attività sulla dBoard; e, infine, di contattare un team remoto per comunicare che l'attività di progettazione è stata completata e cercare informazioni sullo stato di avanzamento del sito remoto. Nel secondo scenario è stata simulata una riunione in piedi; sono stati dati ai partecipanti delle sceneggiature che descrivono i ruoli di vari personaggi (uno dei quali è lo Scrum Master che detta il ritmo del meeting), il lavoro svolto il giorno precedente, quello previsto per il giorno a venire, e se ci fossero degli impedimenti o chiarimenti richiesti dal sito remoto.

Alla fine delle attività sono state poste quattro domande per valutare l'utilità di specifiche funzionalità di dBoard, ad esempio la videoconferenza e lo Scrum board.

Ogni domanda è stata misurata con un punteggio fino a sette punti.

Il workshop ha fornito una panoramica dei risultati del questionario sull'utilità percepita, sulla facilità d'uso e l'utilità delle diverse caratteristiche della dBoard.

Utilità percepita

Anche se nel complesso i partecipanti hanno valutato positivamente l'utilità della dBoard. Alcuni di loro, però, hanno riferito di essere incerti sulla capacità della dBoard di velocizzare il lavoro.

Le stesse perplessità sono emerse anche nelle successive domande relative alla capacità del sistema di: migliorare le prestazioni, migliorare la produttività, migliorare l'efficacia sul lavoro. Alla domanda relativa all'utilità generale della dBoard nel loro lavoro, i partecipanti hanno attribuito un punteggio molto positivo.

Facilità d'uso percepita

Tutti i partecipanti sono riusciti ad appropriarsi del sistema in un brevissimo periodo di tempo e riportato nel questionario che è stato facile diventare abili nell'uso della dBoard.

Nello specifico hanno riferito come sia molto facile imparare ed utilizzare lo strumento. I partecipanti hanno apprezzato l'analogia con i post-it, spesso usati nelle bacheche Scrum fisiche e la flessibilità fornita dalla dBoard in termini di organizzazione libera dei compiti in posizioni arbitrarie. Tuttavia, alla domanda relativa a quanto fosse facile controllare lo strumento, non sono stati assegnati punteggi molto alti.

Utilità delle funzioni

Di fronte a domande più specifiche sull'utilità delle diverse funzionalità fornite dalla dBoard, i punteggi sono risultati molto positivi; soprattutto per quanto riguarda le capacità di videoconferenza, tutti i partecipanti hanno ritenuto questa funzionalità estremamente utile.

Anche le funzionalità della bacheca Scrum sono state accolte molto bene. Alcune lacune, però, sono emerse nella parte relativa alla privacy.

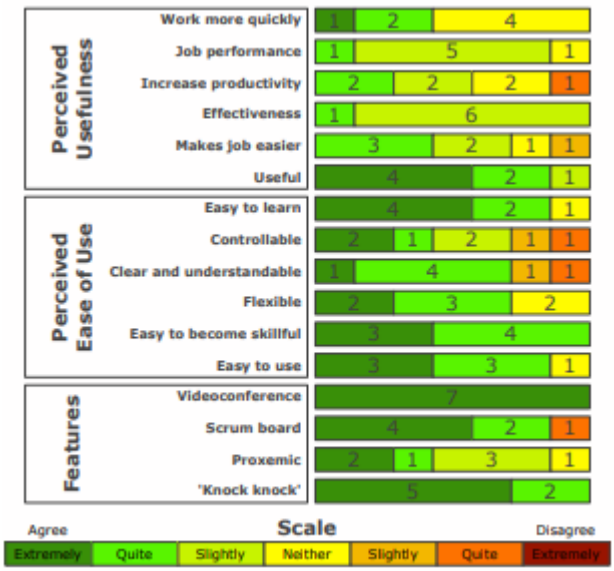

Fig.3– Risultati workshop sulla dBoard.

Discussione

Gli autori dell'articolo, attraverso questo sondaggio, hanno potuto trarre delle conclusioni sull'utilizzo dello strumento. I risultati hanno rivelato che gli utenti hanno trovato la dBoard utile e facile da usare. In particolare, la combinazione fra videoconferenza e Scrum board è stata accolta molto bene.

- **Scaling Scrum**

Nell'articolo di Paasivaara et al. (2012) viene presentato un singolo caso di studio sull'utilizzo dello "Scaling Scrum" in un ampio progetto di sviluppo software distribuito su quattro siti. I dati sono stati raccolti da 19 interviste del personale di progetto, inclusi manager, architetti, sviluppatori e tester.

Nei due anni precedenti alle interviste il progetto era cresciuto di dimensioni ultimi, da due team Scrum collocati a 20 team situati in quattro paesi e che impiegavano oltre 170 persone. Le interviste erano relativamente poco strutturate e colloquiali al fine di mantenere l'adattabilità ai ruoli e alle esperienze individuali dei dipendenti in ruoli diversi. Agli intervistati è stato chiesto di descrivere le proprie esperienze nell'utilizzo di Scrum e i successi e le sfide sperimentate nel progetto relativi all'utilizzo di Scrum. Successivamente è stata organizzata una sessione di feedback in cui i partecipanti erano tutti gli intervistati e anche altri elementi interessati al progetto. La sessione di feedback ha mostrato come l'utilizzo di Scrum in questo grande progetto sia stato molto più controverso di quanto i ricercatori si fossero resi conto durante le interviste. Anche se una parte più ampia del personale del progetto ha trovato questo cambiamento come un passo verso il miglioramento, alcune persone si sono opposte chiaramente al cambiamento. Pertanto, la discussione si è concentrata principalmente sull'utilizzo di Scrum. L'organizzazione del caso è una grande azienda globale di sviluppo. La maggior parte dei progetti di sviluppo prodotto che esegue sono distribuiti a livello globale e coinvolgono sia lo sviluppo software che hardware. In precedenza, l'azienda utilizzava un tradizionale cancello scenico, modello a cascata ma negli ultimi anni l'azienda ha iniziato ad adottare metodi agili per lo sviluppo del software. Il progetto del caso studiato è stato uno dei primi grandi progetti che hanno utilizzato Scrum. La

pratica utilizzata per il ridimensionamento dei meccanismi di coordinamento è l'Area Product Owners.

Area Product Owners (APO)

L'idea era che ogni area di prodotto avrebbe avuto un paio di team di sviluppo e ogni funzionalità sarebbe stata implementata da un team specifico. Tuttavia, a causa del programma serrato, alcune funzionalità sono state suddivise tra diversi team che hanno lavorato su di esse contemporaneamente. La collaborazione e la comunicazione con i team sono state impegnative e non hanno funzionato bene. Le difficoltà sono state causate dalle scelte effettuate dalle APO che hanno la facoltà di designare le squadre per l'implementazione di ogni nuova funzionalità: alcune designazioni non si sono realizzate poiché i team migliori sono stati richiesti da più APO. Sono stati implementati alcuni miglioramenti, sono stati organizzati dei workshop sui requisiti e dei workshop di progettazione su ciascuna user story prima della pianificazione dello sprint, secondo necessità. I workshop sono stati considerati molto utili, questa comunicazione quotidiana tra gli APO e le squadre è stata considerata abbastanza buona.

Discussione

Anche se questo progetto presentava ancora molte sfide sia nell'implementare le pratiche e le idee di base di Scrum, sia nel ridimensionarle, la maggior parte degli intervistati aveva solo cose positive da dire su questo cambiamento di processo, se confrontato con il tipo di modello di processo a cascata utilizzato nei progetti precedenti. L'impresa ha creato una struttura organizzativa abbastanza ben funzionante. Soprattutto il ruolo degli APO è stato considerato molto utile. La crescita rapida dell'organizzazione è stata gestita.

CAPITOLO 5: Considerazioni

Sulla base delle analisi effettuate, estraiamo le seguenti riflessioni:

1) Evitare di distribuire il progetto se non è necessario. Un progetto distribuito comporta ulteriori sfide, maggiore impegno e può richiedere più tempo rispetto a un progetto collocato.
2) Inizialmente è necessario un adeguato allenamento Scrum se il team non ha mai avuto l'occasione di utilizzarlo prima. Oltre ai corsi di formazione in aula, ad esempio, un ingegnere in visita o un esperto esterno potrebbero aiutare il team ad applicare le pratiche Scrum durante la prima iterazione.
3) Scrum si basa su una comunicazione frequente e aperta, e ciò rappresenta il suo principale vantaggio quando applicato a un progetto distribuito. La comunicazione aperta va incoraggiata e garantita in qualsiasi circostanza, altrimenti i benefici di Scrum non possono essere realizzati.
4) Organizzare un accesso a molteplici strumenti di comunicazione, come videoconferenze regolari per riunioni, condivisione desktop e chat per riunioni, può portare notevoli benefici. Strumenti di facile accesso e utilizzo abbassano la soglia per comunicare al di fuori delle riunioni ufficiali.

L'analisi conferma l'applicabilità di Scrum ad una varietà di contesti diversi (dai piccoli team standard co-localizzati allo sviluppo distribuito su larga

scala), ma evidenzia anche la necessità di buone pratiche per garantire che i vantaggi dell'utilizzo di un metodo agile rimangano in vigore. Scrum fornisce una base per la realizzazione di software, ma la metodologia deve essere adattata ad ogni progetto per rispondere a specifiche esigenze. Il modo in cui viene implementato è il fattore chiave che porta al successo o al fallimento del progetto.

Bibliografia e sitografia

- Hron M., Obwegeser N. (2022)
  "Why and how is Scrum being adapted in practice: A systematic review"
  https://www.sciencedirect.com/science/article/abs/pii/S0164121221002077?via%3Dihub

- Vallon R., da Silva Estácio B.J., Prikladnicki R., Grechenig T. (2018)
  "Systematic literature review on agile practices in global software development"
  https://www.sciencedirect.com/science/article/abs/pii/S0950584917302975?via%3Dihub

- Jalali S., Wohlin C. (2010)
  "Agile practices in global software engineering - A systematic map"
  https://ieeexplore.ieee.org/document/5581553

- Esbensen M., Tell P., Cholewa J.B., Pedersen M.K., Bardram J. (2015)
  "The dBoard: A digital scrum board for distributed software development"
  https://dl.acm.org/doi/10.1145/2817721.2817746

- Sungkur R.K., Ramasawmy M. (2014)

  “Knowledge4Scrum, a novel knowledge management tool for agile distributed teams”

  https://www.emerald.com/insight/content/doi/10.1108/VINE-12-2013-0068/full/html

- Lee S., Yong H.-S. (2010)

  “Distributed agile: project management in a global environment”

  https://link.springer.com/article/10.1007/s10664-009-9119-7

- Paasivaara M., Lassenius C. (2011)

  “Scaling scrum in a large distributed project”

  https://ieeexplore.ieee.org/document/6092589

- Paasivaara M., Lassenius C., Heikkilä V.T. (2012)

  “Inter-team coordination in large-scale globally distributed scrum”

  https://dl.acm.org/doi/10.1145/2372251.2372294

- Paasivaara M., Durasiewicz S., Lassenius C. (2009)

“Using scrum in distributed agile development: a multiple case study”

https://ieeexplore.ieee.org/document/5196933