

SCRUM

| Agile Software
| Development





GREAT

BUSINESS

COME FROM GREAT PLANS.

SCRUM

INDICE DEI CONTENUTI

03

Realizzazione di un software

05

Agile Software Development

05

L'innovazione della progettazione Agile

07

Il Framework Scrum.

08

Alla base del Framework Scrum

09

Caratteristiche di Scrum

10

Stakeholder, Team Scrum e i ruoli dei programmatori

11

Team di Sviluppo

12

Product Owner

13

Scrum Master

15

Lo sviluppo del lavoro Scrum: gli Sprint

16

Eventi Scrum

17

Sprint Planning

18

Daily Stand-up

20

Sprint Review

20

Retrospective Meeting

22

Meeting "aggiuntivi": Scrum of Scrums e Product Backlog Refinement

24

Definition of Done

25

Artefatti: quelli principali e gli indicatori di avanzamento

26

Product Backlog

26

Items

27

Sprint Backlog

27

Incremento di Prodotto (o Increment)

28

Task board

28

Burndown chart

29

Burnup chart

29

Cumulative flow

31

Perché scegliere Scrum

33

Ricapitolando

34

Il “manuale operativo” di Scrum

Realizzazione di un software

Un software è un insieme di programmi (procedure ed istruzioni) implicati in un sistema di elaborazione di dati informatici. Esistono diverse tipologie di software: software di sistema, software di base e quello applicativo. Tale distinzione suddivide solo le “superclassi” in cui può essere definita una tipologia di software senza tener conto di una più complessa e approfondita classificazione che avremmo in base ad utilità, caratteristiche, funzionalità ed altri parametri.

Tipi speciali di software si trovano, tuttavia, nei più disparati dispositivi di uso comune (come televisori, elettrodomestici, automobili ecc) che sono ormai diventati parte integrante della nostra vita quotidiana.

È in questo senso che il “creatore” del software trova la sua fondamentale importanza. Un software viene elaborato e realizzato da professionisti esperti in linguaggi di programmazione, i programmatori. Questi gestiscono il processo di sviluppo del software scegliendo tra i diversi modelli di sviluppo esistenti che sono stati codificati dall'ingegneria informatica. Diversi sono i “modelli di sviluppo” e i metamodelli possibili.

Il primo è il modello tradizionale e classico che fa parte delle metodologie definite “pesanti” (Heavyway) ed è quello chiamato waterfall, cioè a cascata. Questi modelli sono sequenziali e quindi il lavoro viene organizzato a partire da un progetto preciso che viene sviluppato stimando costi e tempi in base a ciò che si deve produrre (il prodotto finale da rilasciare o deliverable). In presenza di requisiti ben definiti e documentati, l'approccio di questo modello è adatto. Tuttavia se si ha a che fare con progetti più complessi risulta insufficiente poiché i tempi diventano eccessivamente lunghi, i costi consequenzialmente aumentano e, date le specifiche mutevoli a cui non riuscirebbe a far fronte, il prodotto finale avrebbe una più alta probabilità di fallimento.



UN SOFTWARE PUÒ ESSERE REALIZZATO ATTRAVERSO UNA DELLE TRE TIPOLOGIE ESISTENTI DI “MODELLI DI SVILUPPO”:

SEQUENZIALI
INCREMENTALI
EVOLUTIVI

Per superare tali limiti, il modello a cascata ha subito diversi cambiamenti che hanno portato ai cosiddetti modelli evolutivi. Si tratta in questo caso di modelli di sviluppo ideati successivamente a quelli a cascata.

Una ridotta fase di prototipizzazione venne aggiunta al tradizionale modello a cascata per ottenere dei requisiti. Successivamente si arrivò al modello trasformatore che approfondisce sempre più l'analisi dei requisiti al fine di rilasciare un prototipo funzionale.

Successivamente ai modelli evolutivi venne abbinata un'accezione di lavoro incrementale. Fu così che nacque quindi il modello a spirale, definito come un metamodello iterativo che, come dice la parola stessa, utilizza le iterazioni (un blocco o ciclo di operazioni) che rende più semplice la verifica del lavoro prodotto e quindi l'eventuale adattamento a ciò che il cliente richiede. Questo modello è in effetti un modello incrementale ed è definito metamodello perché si basa sulla gestione del rischio utilizzando lo sviluppo a cascata ma in iterazioni alla fine delle quali vengono rilasciati dei prototipi da presentare al cliente. In questo modo si riduce il rischio di fallimento.

Ancora il modello a fontana è altamente iterativo e ideato per lo sviluppo di progetti che prevedono la collaborazione con un gran numero di persone e definiti "mission critical".

Tuttavia il più recente modello di progettazione di un software è la metodologia agile di cui fa parte il processo Scrum. Si tratta una metodologia evolutiva che utilizza l'approccio iterativo e si concentra su tempi e costi concordati con il cliente gestendo il lavoro in base alle esigenze di rapidità. Eventuali necessità di modifiche vengono gestite all'occorrenza attraverso le versioni del prodotto da rilasciare.

I modelli agili si contrappongono al classico modello a cascata e agli altri modelli tradizionali in quanto il lavoro si focalizza sull'obiettivo da raggiungere ma in tempi più brevi e rilasciando frequentemente funzionalità del prodotto per realizzare un progetto complesso e di alta qualità.

Tale approccio prevede infatti la possibilità di modifiche o aggiungere elementi man mano che la committenza ne dichiara la necessità, senza doverle prevedere tutte fin dall'inizio dell'avvio del progetto.

Questo è il motivo per cui le metodologie agili sono più indicate nello sviluppo di progetti complessi quali lo sviluppo software o ricerca e fasi di ingegnerizzazione.

Agile Software Development

IL MINDSET AGILE: UNA FAMIGLIA DI METODI LEGGERI

La metodologia agile (ASD ossia Agile Software Development), è un mindset (un paradigma organizzativo) costituito da un insieme di metodologie per lo sviluppo di un software.

Le metodologie agili sono anche definite con il termine Lightway (leggero), proprio in contrapposizione ai metodi classici definiti invece pesanti (Heavyway). Diversi sono i metodi (e i framework) che condividono comunemente la caratteristica di leggerezza e flessibilità propria dello sviluppo agile, e sono: Agile Unified Process, Adaptive Software Development, Crystal, Dynamic System Development Method, Extreme Programming, Feature Driven Development, Lean software Development e il Framework Scrum. Queste metodologie sono state ideate per ottimizzare le varie fasi dello sviluppo e per rendere al cliente un prodotto di qualità, riducendo i tempi ed i costi. Il mindset agile deriva dal “paradigma di autonomia condivisa”, in cui i professionisti coinvolti nel team collaborano tra loro scandendo le iterazioni come progetti singoli che sviluppano gradualmente le funzionalità del software stesso.

Tali pratiche promuovono la formazione di team di sviluppo polifunzionali capaci di auto-organizzarsi per condurre lo sviluppo del progetto adottando un approccio iterativo ed incrementale. I metodi agili tendono infatti a minimizzare le possibilità di fallimento organizzando il lavoro in finestre, chiamate iterazioni. Ogni iterazione, dalla durata limitata nel tempo (da una a quattro settimane con una preferenza per periodi più brevi), è un progetto che viene pubblicato e rilasciato, rappresentando uno “step” del lavoro complessivo. In questo modo si va incontro al cliente poiché quest’ultimo viene coinvolto direttamente dal team di lavoro che pianifica di volta in volta le varie fasi, in maniera adattiva poiché si basa sulle richieste ed i feedback che raccoglie al termine di ogni fase. Così facendo si valutano man mano le priorità del progetto e si possono integrare eventuali modifiche, per soddisfare le richieste del cliente.

L’innovazione della progettazione Agile

I VALORI DELL’INNOVAZIONE AGILE

Nel 2001 Kent Beck, Robert C. Martin, Martin Fowler e i loro collaboratori, pubblicarono il “Manifesto per lo sviluppo agile del software”, con l’intento di definire i valori e i principi di una nuova metodica che si contrapponesse ai precedenti metodi obsoleti. Questi dovevano essere superati perché risultavano troppo lenti e costosi e spesso incapaci di portare a termine un progetto che soddisfacesse a pieno le richieste del cliente.

Una grande innovazione che cambiò totalmente il modo di progettare, spostando l’ago della bilancia su altre priorità.

Di seguito vengono identificati e descritti i valori su cui si basano queste metodologie innovative.

- Gli individui e le interazioni sono più importanti degli strumenti e dei processi di progettazione. In questo senso il cliente si affida totalmente al team di sviluppatori che provvede alla realizzazione del progetto e al superamento dei problemi e all'introduzione di eventuali modifiche relativi ad esso e che si presentano durante lo svolgimento del lavoro. Il cliente riveste un ruolo fondamentale, venendo invitato direttamente dal team che discute con lui i risultati di ogni iterazione al fine di eliminare ogni possibilità di fallimento del prodotto finale.

- Al di là dell'eshaustività della documentazione si mira alla funzionalità
- del software che regola automaticamente il lavoro, guidando il progetto fino al suo successo. Ovviamente è prevista una documentazione con relativo lavoro di analisi, test ecc, tuttavia per l'avanzamento vero e proprio del progetto non si fa più fede a questa o più in generale ad un piano iniziale, quanto al compimento delle varie fasi (chiamate sprint) che sono le iterazioni in cui è diviso il lavoro del progetto. Queste, svolte come singoli progetti, forniscono prodotti "finiti" da poter rilasciare al cliente e su cui poter proseguire il lavoro complessivo che viene gestito e implementato in base alla loro riuscita.

- La collaborazione con il cliente è il punto di forza per la realizzazione del progetto che non si basa più sulla sterile negoziazione del contratto.
- Secondo questo principio della metodologia agile, ci si assicura che il prodotto finale abbia il più alto valore qualitativo possibile, facendo in modo da riscontrare questa caratteristica anche nelle varie fasi di sviluppo del prodotto.

- Le linee guida per la progettazione stessa derivano dalla capacità di rispondere a gradualmente cambiamenti direzionati dalle richieste del
- cliente invece che da un "piano immutabile". Il progresso del progetto, quindi, è dato dall'incremento del prodotto che viene garantito per ogni sprint che si porti a termine.

Il controllo dello sviluppo si sposta così a livello operativo, guidato da scopi e misure che vengono definite dal cliente stesso e affrontate dal team di sviluppatori che ispezionano continuamente il loro lavoro adattandolo alla realtà a cui si mira. La tipologia di lavoro si focalizza quindi su una logica pull redatta tramite la successione graduale di step eseguiti in un tempo limitato da team multifunzionali. Questi sono auto-organizzati e quindi responsabili end-to-end delle singole iterazioni del lavoro svolto mirando alla User Experience più che ad ultimare il prodotto finale.

Il Framework Scrum

Tra le varie metodologie di sviluppo raggruppate con il termine “Agile” e proposte agli inizi del 2000, Scrum è di gran lunga il più diffuso e da più tempo utilizzato per la realizzazione di prodotti complessi.

Si tratta di un framework, un processo utilizzato per la gestione del ciclo di vita (e di sviluppo) del software concepito specificatamente per progettare o realizzare tale prodotto o applicazioni di sviluppo.

Un framework è generalmente, una struttura logica di supporto utilizzata per facilitare le fasi di lavorazione dello sviluppo. Esso rappresenta quindi un particolare design pattern (ossia una soluzione generale per un problema ricorrente nell’ambito della progettazione), che facilita quindi il lavoro di realizzazione del prodotto.

Nel 1986 furono Hirotaka Takeuchi e Ikujiro Nonaka a gettare le basi per la creazione del framework Scrum. Essi descrissero un innovativo modo di sviluppare prodotti commerciali, concentrandosi principalmente sulla velocità e flessibilità durante i processi di lavorazione. Il loro studio aveva interessato casi idealizzati sull’industria automobilistica e quella di fotocopiatrici e stampanti. Lo identificarono come approccio olistico perché eseguito da un team interfunzionale che lavorava per più fasi fino al raggiungimento di un obiettivo comune.

Già dal 1990 era quindi, impiegato nello sviluppo di software di vario genere e hardware e per reti di funzioni interagenti usati quotidianamente da individui e società in vari ambiti: nelle scuole, nel mondo del marketing, nel governo, insomma ovunque potesse essere impiegato. I suoi ideatori furono Ken Schwaber e Jeff Sutherland che svilupparono Scrum per agevolare i programmatori nella risoluzione dei problemi più diffusi ossia quelli adattivi rispetto alle esigenze del cliente e alla necessità di sviluppo di prodotti complessi ed efficienti.

Venne però presentato e documentato ufficialmente nel 1995 durante la conferenza OOPSLA. Furono i suoi stessi creatori a definirlo: “leggero e semplice da comprendere, tuttavia difficile da padroneggiare”.

L’efficace funzionalità di Scrum risiede nel fatto che questo framework ha alla base della sua operatività un concetto continuo di miglioramento. Scegliere Scrum significa mirare ad ottimizzare il prodotto stesso durante tutte le fasi necessarie al suo sviluppo, migliorare il lavoro del team di sviluppatori ed evolvere l’ambiente lavorativo ad un livello di cooperazione e complicità unica.



SCRUM: IL FRAMEWORK
DEL MODELLO AGILE PER GESTIRE
IL CICLO DI VITA DEL SOFTWARE.
IL PROCESSO SCRUM OTTIMIZZA
IL PRODOTTO, MIGLIORANDONE
LA LAVORAZIONE.

Alla base del Framework Scrum

I PILASTRI SU CUI SI BASA IL PROCESSO SCRUM:

- TRASPARENZA
- ISPEZIONE
- ADATTAMENTO

Per poter padroneggiare Scrum è necessario anzitutto, tener presente i tre principi, i pilastri fondamentali su cui esso è basato: trasparenza, ispezione ed adattamento.

Secondo il primo pilastro che è quello della Trasparenza, si intende che tutti gli aspetti più significativi, riguardanti il lavoro di sviluppo del software, devono essere ben chiari ai responsabili del progetto. Tale chiarezza (e quindi trasparenza), è data dall'utilizzo di standard comuni per una comune comprensione di ciò a cui si mira e quindi ad una visione complessiva del prodotto di volta in volta realizzato, attraverso il proprio lavoro e la collaborazione con gli altri componenti del team che ne condividono una comune comprensione.

Tale principio non può non essere connesso a quello dell'Ispezione. Come tutte le metodologie agili, il framework Scrum prevede di ispezionare continuamente i singoli prodotti delle iterazioni e analizzando e valutando i relativi artefatti in modo da individuare eventuali difformità rispetto allo Sprint Goal cui vuole volgere l'avanzamento del lavoro. Occasione di ispezionare il lavoro svolto viene offerta al cliente, attraverso il rilascio delle funzionalità prodotte in ogni iterazione, e ad ognuno dei componenti del team che ispeziona continuamente il lavoro completato o da produrre. Coloro che ispezionano il lavoro del Team Scrum, quindi, rispettano il terzo ed ultimo principio di Scrum relativo all'Adattamento.

L'ispezione ha appunto l'obiettivo di adattare quanto è più possibile i singoli prodotti delle varie fasi dello sviluppo ad un prodotto finale idealizzato dal cliente e che soddisfacesse tutti i requisiti da lui desiderati. Se si rilevano aspetti imprevisti che vanno oltre i limiti di accettabilità del prodotto, bisogna intervenire prontamente per ridurre al minimo l'errore e quindi il fallimento del prodotto finale. Ispezione ed adattamento vengono eseguiti in precise occasioni formali previste dallo svolgimento del lavoro che sono definite dai vari "eventi Scrum", mentre la trasparenza è verificata anche con la produzione degli Artefatti.

Caratteristiche di Scrum.

SCRUM ORGANIZZA IL LAVORO DA SVOLGERE, DIVIDENDO IL LAVORO IN CICLI SPRINT. IL TEAM SCRUM, COME UNA SQUADRA DI RUGBY, SI AUTO-ORGANIZZA. ATTRAVERSO DEGLI EVENTI PREVISTI DAL PROCESSO E LA PRODUZIONE DI ARTEFATTI, INCREMENTA GRADUALMENTE IL LAVORO DA REALIZZARE

La metodologia Scrum prevede di dividere il lavoro in blocchi o step singoli, identificati con gli Sprint, che costituiscono le varie iterazioni del processo di sviluppo di un software. Uno Sprint si ripete ciclicamente durante il processo Scrum, al fine di portare al completamento di una o più funzionalità previste per il prodotto finale. Per questo motivo si parla di più cicli Sprint che costituiscono quindi le fasi di sviluppo del processo. Ognuna di queste fasi, è limitata nel tempo (time boxed) e deve portare, quindi, alla produzione di parti di prodotto funzionanti che possano essere immediatamente rilasciate al cliente. Scansionando il tempo di lavorazione in periodi più brevi e limitati, si gettano le basi che costituiscono le fondamenta di un processo di lavorazione in cui tutto deve essere ottimizzato e non è tollerabile lo spreco di tempo.

Il processo Scrum, quindi, è una metodologia che prevede la divisione dello sviluppo di un software in base ad un tempo prefissato e limitato, in fasi di lavorazione (organizzate nei tempi più brevi possibile), in cui ci si dedica ad “un pezzo per volta” del prodotto finale, con lo scopo di incrementare gradualmente il lavoro.

Questo processo potrebbe essere associato all'immagine della lavorazione in una qualsiasi fabbrica. Tuttavia il lavoro non viene svolto da macchinari e robot, ma da una squadra di sviluppatori e professionisti che aspira a migliorare le proprie interazioni. Il lavoro che ogni componente di questa squadra svolge, deve essere coordinato e funzionale a quello degli altri per poter incrementare la lavorazione fino al raggiungimento di un unico obiettivo comune a tutto il gruppo.

Scrum, come del resto anche le altre metodologie agili, si basa sul paradigma organizzativo ed è quindi il concetto stesso di organizzazione del lavoro ad essere la parola chiave di tale processo. Ciò si può espletare solo attraverso la pianificazione di specifici incontri (i meeting che costituiscono i vari eventi previsti durante il processo Scrum), che sono le occasioni per mettere in chiaro cosa e come si sta procedendo nella lavorazione. Solo in questo modo è possibile ispezionare il lavoro stesso per adattarlo all'obiettivo finale. Si tratta quindi di un processo che si basa fortemente sull'operatività del Team di lavoro.

Il nome “Scrum” con cui si identifica questo processo di sviluppo, fa riferimento al termine utilizzato nel gioco del rugby. Letteralmente si traduce in italiano come “mischia” ed indica una precisa disposizione dei giocatori di rugby che si posizionano sul campo assumendo appunto l'aspetto di una “mischia chiusa o ordinata”.

Tale disposizione viene adottata in una fase del gioco che si crea durante la ripresa e successivamente all'interruzione dovuta al verificarsi di un'irregolarità. Così come nel gioco del rugby, in un processo Scrum, il lavoro viene svolto da una squadra interfunzionale seguendo più fasi in modo da raggiungere l'obiettivo, agendo come fosse un'unica entità coordinata. La metodologia Scrum si può definire, quindi, un processo fondato sull'empirismo perché proprio come questa filosofia, le decisioni si basano su ciò che si conosce e a sua volta la conoscenza deriva dall'esperienza.

Applicando lo stesso principio dell'empirismo alla lavorazione con Scrum, ogni fase dello sviluppo (gli script) durante la quale si lavora ad una funzionalità del software da sviluppare, costituisce l'esperienza del lavoro che deve portare ad un prodotto rilasciabile perché considerabile "fatto" o "done". Al termine di ogni ciclo Sprint, il Team Scrum volge l'attenzione su ciò che ha prodotto e le modalità di lavorazione (quindi l'esperienza) e pianifica la prossima fase del processo, basandosi proprio sui feedback che guadagna (e che determinano quindi la qualità del lavoro svolto).

Stakeholder, Team Scrum e i ruoli dei programmatori

GLI INDIVIDUI COINVOLTI NEL PROCESSO SCRUM SI DISTINGUONO TRA STAKEHOLDER E TEAM SCRUM.

IL TEAM OPERATIVO PREVEDE TRE RUOLI:

IL TEAM DI SVILUPPO, IL PRODUCT OWNER E LO SCRUM MASTER.

Originariamente, tutte le persone coinvolte durante la lavorazione di un prodotto, condotto con il metodo Scrum, venivano raggruppati e divisi in due insiemi distinti che assumevano il nome di: "maiali e polli". Tale distinzione prendeva spunto dalla fiaba "The chicken and the pig" utilizzandola come metafora per definire il diverso tipo di approccio ad un progetto, in base al livello di coinvolgimento.

La favola è spesso riassunta sotto forma di uno spiritoso indovinello che recita così:

"- In una colazione a base di bacon e uova, qual è la differenza tra il Pollo ed il Maiale?

- Il Pollo è interessato, ma il Maiale è coinvolto!"

Secondo questa vecchia metafora quindi, il "pollo" veniva identificato proprio con lo stakeholder, persona (o insieme di persone) che partecipava in modo sporadico al progetto, senza dare un contributo continuo ma dirigendo il lavoro complessivo.

Per intenderci, gli stakeholders sono i soggetti o gruppi che hanno necessità del prodotto e ne commissionano quindi la realizzazione. Essi rappresentano il cliente e perciò la parte “interessata” ed influente. Il “maiale”, invece, si comportava come figura contrapposta alla precedente poiché era utilizzata per rappresentare la persona (o il gruppo) che si dedicavano completamente e integralmente al progetto, lavorando con costanza.

Questo gruppo che costituisce il cuore operativo in un processo Scrum, è definito Team Scrum (o semplicemente Team). Successivamente questa divisione tra individui interessati all'interno di uno stesso progetto, venne però superata.

Attualmente, senza applicare nessuna metafora, gli stakeholder vengono semplicemente distinti dal Team Scrum. Tale gruppo prevede tuttavia una diversificazione interna per ruoli, secondo cui, nel processo Scrum si distinguono: il Team di sviluppo, il Product Owner e lo Scrum Master.

Il Team di sviluppo

IL TEAM DI SVILUPPO È IL RUOLO RICOPERTO DALLA SQUADRA DI SVILUPPATORI:

- *DEFINISCONO LO SPRINT BACKLOG*
- *PROCEDONO NELLA LAVORAZIONE DEL*
- *PRODOTTO DI OGNI SPRINT SEGUENDO UNO SPRINT PLAN*
- *PORTANO A TERMINE LA LAVORAZIONE DEL PRODOTTO DEI CICLI SPRINT*

La squadra di sviluppatori che che insieme collaborano tra loro attivamente per realizzare il prodotto nelle sue varie fasi, costituisce il cosiddetto Team di sviluppo (o Dev Team).

Questo gruppo conta generalmente dalle tre alle nove persone per motivi specifici: se il gruppo contasse meno di tre individui si avrebbe una risultante interazione minima e quindi un eccessivo rallentamento nelle tempistiche di produzione. Al contrario, coinvolgere più di nove persone all'interno del Team di Sviluppo, significherebbe complicare eccessivamente ed inutilmente il relativo coordinamento tra “troppi” operatori. Il Team di Sviluppo ha competenze cross-funzionali e sono capaci, quindi, di sviluppare effettivamente ogni aspetto della funzionalità da produrre (rappresentata dall'Incremento di Prodotto) che costituisce l'obiettivo a cui deve portare la lavorazione di ogni ciclo Sprint.

Il lavoro di analisi, progettazione, sviluppo, test così come la comunicazione tecnica e la produzione della relativa documentazione del lavoro, è prodotto dalla squadra degli sviluppatori.

Questi professionisti, quindi, collaborano strettamente gli uni con gli altri auto-organizzandosi al fine di raggiungere l'obiettivo di ogni fase. Un componente di questo gruppo può essere incaricato del ruolo accessorio di Manager per meglio controllare l'ambiente di lavoro. Tuttavia non si tratta di un ruolo formale che è ufficialmente riconosciuto nel processo Scrum ma è solo di un ruolo ausiliario che può essere preso in considerazione oppure no.

L'auto-organizzazione risiede nella responsabilità che compete a questo gruppo che, in base alle proprie esigenze, stabilisce quanto lavoro sarà effettivamente svolto durante ogni Sprint per produrre, al fine di ogni fase, un incremento del prodotto immediatamente rilasciabile ed utilizzabile dal cliente, definito "fatto" o "done". Attraverso il rilascio incrementale, il team di sviluppo aumenta anche il numero di feedback funzionali a questo processo empirico. Seguono quindi uno Sprint Plan (cioè un piano dello sprint) che rispetta una determinata organizzazione stabilita dal Product Owner.

Product Owner

IL RUOLO DEL PRODUCT OWNER DIREZIONA LA LAVORAZIONE DEL PRODOTTO ORGANIZZANDO IL PRODUCT BACKLOG DI CUI È L'UNICO RESPONSABILE. RAPPRESENTA IL TRAMITE TRA L'AZIENDA E IL CLIENTE E IL SUO OBIETTIVO È QUELLO DI OTTIMIZZARE IL LAVORO DEL TEAM SCRUM PER MASSIMIZZARE IL VALORE DEL PRODOTTO FINALE.

Un altro ruolo che direziona parzialmente l'organizzazione che viene seguita dal Team di Sviluppo, è impersonificato nella figura del Product Owner. Egli rappresenta l'intermediario tra gli stakeholders e il Team Scrum ed è quindi responsabile di far produrre un progetto con il più alto valore qualitativo possibile entro la data di termine desiderata dal cliente. Il Product Owner è quindi la "voce del cliente" e si occupa dell'aspetto che riguarda la caratteristica di business che appartiene al progetto. Per questo motivo il PO deve avere una conoscenza approfondita dei bisogni degli utenti finali e capire in che modo il prodotto può essere una risposta a queste necessità.

Deve quindi saper gestire gli stakeholder ed essere un tramite con il Team Scrum, comunicando la più adatta vision di sviluppo al resto del gruppo. Nel fare ciò tuttavia deve avere anche un'approfondita comprensione del mercato e delle capacità della propria azienda per coordinare al meglio il processo di sviluppo.

In questo senso influenza il lavoro del Team di sviluppatori poiché è suo compito scegliere e definire gli item (requisiti) in un Product Backlog che è sostanzialmente una lista di caratteristiche che deve avere il prodotto da sviluppare, elencate secondo un ordine prioritario scelto proprio dal Product Owner.

La figura del Product Owner è l'unica responsabile dell'organizzazione e della gestione del Product Backlog e ha il compito di assicurarsi che tutti i componenti del Team di sviluppo sappiano chiaramente cosa contiene e quali siano le priorità da tener presente durante la lavorazione.

Ogni Team Scrum deve quindi essere coordinato da un Product Owner collaborando con lui.

Il ruolo del Product Owner è una figura chiave del Team Scrum poiché deve riuscire ad instaurare una buona comunicazione sia con il cliente (lo stakeholder) che con il gruppo operativo del Team Scrum. Questo ruolo deve perciò essere ricoperto esclusivamente da un individuo che abbia determinate qualità e caratteristiche, quali disponibilità, doti comunicative e capacità imprenditoriali.

La responsabilità che compete a questo ruolo è in definitiva, quella di “far uscire il prodotto” nel modo che più soddisfi le richieste dello stakeholder, definendo e comunicando con chiarezza quali caratteristiche deve possedere il prodotto finale. Può infine influenzare le tempistiche di produzione scegliendo di dare la priorità ad un aspetto del lavoro più che ad un altro (proponendo quindi in un certo ordine i vari requisiti e funzionalità da sviluppare durante fasi della lavorazione).

Collabora attivamente con il Team di Sviluppo determinando con il suo lavoro i tempi di sviluppo effettivi e la qualità del prodotto finale. L'obiettivo del Product Owner è quello di ottimizzare quanto è più possibile il lavoro, massimizzando, al contempo, il valore del prodotto da rilasciare.

Scrum Master

LO SCRUM MASTER COME LEADER SERVANT DEL TEAM DI SVILUPPO. ISTRUISCE, GUIDA E SUPPORTA LA SQUADRA DEGLI SVILUPPATORI NELL'APPLICAZIONE ED ESECUZIONE DEL PROCESSO SCRUM.

La terza fondamentale figura di un Team Scrum, è quella dello Scrum Master. Anche se lo Scrum Master non è direttamente coinvolto nella lavorazione del prodotto, questa figura possiede comunque un ruolo fondamentale all'interno di un Team Scrum.

Egli aiuta il resto del Team a perseguire gli obiettivi della lavorazione rimuovendo gli ostacoli che, al contrario, limiterebbero la buona riuscita di ogni Sprint. Uno Scrum Master deve quindi conoscere approfonditamente il framework Scrum per poter istruire e guidare gli altri componenti del team nella sua comprensione ed applicazione, fungendo da formatore del gruppo. Inoltre egli conduce e assiste il Team di Sviluppo promuovendo effettivamente la giusta esecuzione del processo Scrum.

Non si tratta del leader del team ma può essere definito come un leader servant poiché si comporta effettivamente come un leader ma si pone al servizio del gruppo poiché ha il compito di facilitare l'applicazione del framework Scrum nelle varie fasi di lavorazione.

Come mediatore, affianca il Product Owner e collabora con lui suggerendogli come organizzare correttamente il Product Backlog a cui il resto del Team dovrà attenersi. Contemporaneamente però, lavora a stretto contatto con il Team di sviluppo individuando e suggerendo alla squadra, le tecniche che possano facilitare le fasi del processo Scrum.

Come un coach, lo Scrum Master conosce perfettamente le norme, le tecniche, le pratiche e i valori di Scrum ed è quindi l'unico che possa rispondere e chiarire qualsiasi domanda o dubbio in merito al processo

Lo Scrum Master si comporta come un "cuscinetto" tra il Team Scrum e l'esterno presiedendo spesso le riunioni importanti che vengono condotte ed organizzate proprio da questa figura e difendendo il Team di Sviluppo da qualsiasi influenza che possa disturbarne il lavoro.

Egli è l'unico esperto della metodologia Scrum e la sua responsabilità è perciò quella di controllare la lavorazione guidando, revisionando e supportando la squadra nella produzione degli artefatti. Lo Scrum Master è il punto di riferimento dell'intera squadra poiché grazie alle sue spiccate qualità di problem solving, è capace di superare ogni ostacolo che possa presentarsi durante il processo.

Direziona quindi il Team a lavorare al meglio, nel modo più produttivo possibile.

La figura dello Scrum Master diventa fondamentale poiché la metodologia Scrum, nonostante sia molto versatile ed è quindi possibile utilizzare questo processo non solo per la realizzazione di software, ma anche di hardware e servizi vari, risulta particolarmente complicata da adottare se non la si conosce a pieno. In un team in cui questo processo non è del tutto compreso, questa figura è l'unica capace di assistere il team nella realizzazione di prodotti complessi di alto valore qualitativo. Egli è il solo che possa rendere l'intero Team più dinamico ed efficiente, applicando nel giusto modo il metodo Scrum.

Senza il suo contributo il processo Scrum non può essere eseguito ed applicato e ciò comporterebbe quindi una più alta probabilità di fallimento del prodotto.

Lo sviluppo del lavoro in Scrum: gli Sprint

L'ORGANIZZAZIONE DEL LAVORO NEL PROCESSO SCRUM PREVEDE LA DIVISIONE DELLA LAVORAZIONE IN CICLI SPRINT. GLI SPRINT SONO TIME-BOXED IMPLEMENTANO IL LAVORO CON IL RILASCIO DI UN INCREMENTO DEL PRODOTTO AL TERMINE DI OGNI ITERAZIONE. OGNI SPRINT È COSTITUITO DA QUATTRO EVENTI SCRUM.

Nel processo Scrum il lavoro è diviso in progetti singoli che devono avere una durata fissa e limitata nel tempo (time-boxed). Ognuno di queste fasi corrispondono ad un'unità fondamentale dello sviluppo e viene definito da tutto il Team durante precisi momenti, le riunioni prestabilite, che costituiscono gli "eventi Scrum".

Il processo Scrum, come tutti i metodi agili, non tiene più conto fedelmente di un "piano" originario e immutabile. Tuttavia ciò non toglie che esiste una certa pianificazione e progettazione del lavoro da seguire. Una linea guida è fornita infatti dal Product Owner attraverso un artefatto fondamentale da lui stilato, quale è il Product Backlog. Questa lista di elementi del lavoro da portare a termine è ordinata in base alle priorità che devono essere implementate durante le fasi dello sviluppo. L'obiettivo da perseguire è sostanzialmente il prodotto finale. Di questo, come in una fabbrica, ci si dedica "pezzo per pezzo" rispetto alle varie funzionalità desiderate dallo stakeholder.

Il lavoro complessivo che viene realizzato in modo incrementale e suddiviso quindi in fasi, è costituito da progetti singoli che devono essere svolti durante le varie iterazioni, che corrispondono ai cicli Sprint.

Dunque lo Sprint può essere definito come una delle fasi, un'unità del lavoro, in cui tutti i componenti del Team Scrum (il Team di Sviluppo, il Product Owner e lo Scrum Master), collaborano tra loro per creare, modificare e migliorare il prodotto attraverso la produzione ed il rilascio di sue parti funzionali (e funzionanti).

L'iterizzazione attraverso gli Sprint è continua e incrementale e ognuno di questi cicli porta a termine unità di lavoro attraverso il rilascio di un Incremento di Prodotto. Questo è un altro artefatto che viene ispezionato e testato per poter essere rilasciato al cliente solo se soddisfa i criteri di accettazione che vanno sotto al nome di Definition of Done.

Portato a termine uno Sprint si inizia subito con il successivo.

Ogni Sprint, viene preceduto, guidato e seguito da precisi eventi previsti dal processo Scrum. Questi sono le quattro tappe che caratterizzano ogni fase dello sviluppo e si differenziano l'uno dall'altro, in base alla funzione a cui assolvono.

Eventi Scrum

IL PROCESSO SCRUM PREVEDE DI SCANDIRE OGNI CICLO SPRINT ATTRAVERSO L'ORGANIZZAZIONE DI SPECIFICI EVENTI:

- *SPRINT PLANNING*
- *DAILY STAND-UP*
- *SPRINT REVIEW*
- *RETROSPECTIVE MEETING*
- *PRODUCT BACKLOG REFINEMENT ED EVENTUALMENTE SCRUM OF SCRUM*

Un processo Scrum è una metodologia in cui il lavoro complessivo viene eseguito implementando piccoli comparti, gli Sprint, in cui ci si dedica ad una delle funzionalità che deve possedere il prodotto finale. Ogni Sprint però è pianificato, definito, revisionato e migliorato attraverso degli incontri. Si comporta quindi come un ciclo formato da quattro (o più) eventi ben definiti in ognuno dei quali si rispecchiano i tre principi fondamentali del processo Scrum. Innanzitutto questi meeting servono a chiarire l'obiettivo a cui mira il lavoro da svolgere e quindi rispetta a pieno il principio di Trasparenza che sussiste tra il Team Scrum e il cliente (ma anche all'interno dei componenti del gruppo).

Inoltre ognuno degli eventi costituisce un'occasione formale per ispezionare l'iterazione prodotta e adattarla al lavoro da produrre (soddisfacendo così anche i principi di Ispezione ed Adattamento).

Gli eventi Scrum sono come dei precisi rituali e vengono ripetuti con regolarità in modo da assicurare che sia spesa un'opportuna quantità di tempo, senza ulteriori e imprevisti sprechi durante le fasi del processo. Tuttavia è fondamentale rispettare categoricamente la frequenza di questi incontri perché essi, aumentando il numero di feedback del lavoro svolto, servono a minimizzare le possibilità di fallimento del prodotto definitivo.

Ogni ciclo Sprint quindi, consiste di quattro specifici eventi.

Lo Sprint ha inizio con lo Sprint Planning, in cui si pianifica il progetto relativo al singolo Sprint, organizzando e stilando uno Sprint Backlog che ne guidi la lavorazione.

Il Team, poi, si auto-organizza di volta in volta per sviluppare l'Incremento di Prodotto, attraverso il Daily Stand-up (o Daily Scrum). Questo incontro è funzionale al coordinamento del gruppo. Seguono infine lo Sprint Review e il Retrospective Meeting (o Sprint Retrospective) che si tengono una volta portato a termine lo Sprint per poter analizzare e revisionare il prodotto.

Uno Sprint successivo verrà organizzato preparando il prossimo Sprint Planning durante un meeting chiamato Product Backlog Refinement.

Sprint Planning

UNO SPRINT INIZIA CON LO SPRINT PLANNING, EVENTO IN CUI VIENE PIANIFICATO ED ORGANIZZATO IL LAVORO DA SVOLGERE NEL CICLO SPRINT. IL DEV TEAM PRODUCE LO SPRINT BACKLOG E DETERMINA LA DURATA DELLO SPRINT.

Per dare inizio al lavoro che verrà svolto in uno Sprint, bisogna pianificare ciò che deve essere eseguito e stimare anche la durata di lavorazione che questo lavoro prevede. Tutto deve rientrare infatti, in tempi ben definiti, così ogni fase, evento e lavorazione, nel metodo Scrum è time-boxed (come suggerisce la parola è inscatolato in un lasso di tempo).

Ciò viene deciso in un primo incontro, della durata massima di due ore per ogni settimana in cui si protrae il lavoro dello Sprint (quindi l'incontro può essere di otto ore massimo), chiamato Sprint Planning. Attraverso questo "piano di attacco" il Team di Sviluppo definisce il lavoro che dovrà essere svolto nel corso dello Sprint.

Attinge e seleziona gli item che verranno effettivamente eseguiti in questo ciclo, dalla lista del Product Backlog fornita dal Product Owner. Il Product Owner, che fa da mediatore tra il gruppo operativo ed il cliente, dà inizio all'incontro comunicando a tutto il Team Scrum quali elementi prioritari stilati nel Product Backlog, dovrebbero essere completati con più urgenza. Tutti devono comprendere il lavoro che deve essere svolto, tuttavia tra gli elementi proposti dal Product Owner, solo ed esclusivamente il Team di Sviluppo determina quali e quanti possono essere effettivamente presi in carico e completati durante il prossimo ciclo Sprint. Tale scelta viene temperata in base a diversi aspetti: ad esempio tenendo conto delle passate performance del gruppo (qualora non fosse il primo Sprint), oppure dell'attuale stato di avanzamento del prodotto o ancora delle capacità proprie degli sviluppatori.

Insomma è il Team di Sviluppo a dover eseguire il lavoro quindi solo questo gruppo può sapere quanto tempo impiega a portare a termine la lavorazione. Una volta definiti gli item del prodotto su cui lavorare, viene organizzato un artefatto: lo Sprint Backlog.

Questo è esclusivamente di proprietà del Team di Sviluppo e quindi nessun altro componente del Team Scrum può modificarne il contenuto. Come il Product Backlog è una lista di elementi da completare, ma relativa solo all'imminente ciclo Sprint. Lo Sprint Backlog è assimilabile come concetto, ad un piano sufficientemente dettagliato, utile al Team di Sviluppo per garantire il risultato desiderato da ogni Sprint. Viene poi deciso anche la durata prevista per sviluppare tale lavoro e quindi di quante settimane è composto lo Sprint. Non è possibile comunque oltrepassare il limite massimo di quattro settimane.

Spesso ad uno Sprint viene assegnato anche un Sprint Goal che faccia focalizzare gli sviluppatori sulla meta da raggiungere attraverso la lavorazione, più che sui relativi dettagli. Si tratta di un obiettivo che deve essere raggiunto attraverso il lavoro dello Sprint a cui viene associato e che sarà raggiunto producendo un Incremento di Prodotto consono agli item prescritti nel Product Backlog. Fornisce quindi indicazioni fondamentali al Team di Sviluppo sul motivo per cui deve svolgere il lavoro dello Sprint riducendo le “distrazioni” in cui possano durante la lavorazione.

Il Team degli sviluppatori, definisce in questa fase anche le modalità con cui porteranno a termine il lavoro. Analizzando e progettando il lavoro da svolgere durante lo Sprint, infatti, è possibile che venga deciso di suddividere ulteriormente il progetto in parti più piccole e giornaliere. Queste sono definite “storie” (user story) ed hanno la durata di un giorno o anche meno.

Il Product Owner oltre a proporre gli elementi del Product Backlog che potrebbero essere presi in carico nell'imminente Sprint dal Team di Sviluppo, chiarisce sulle caratteristiche e i requisiti che ci si aspetterebbe dall'Incremento di Prodotto su cui si sta per intraprendere il lavoro.

Una volta determinato indirettamente il “come” dal Product Owner e il “quanto” dal Team di Sviluppo, l'incontro può essere considerato concluso, a patto però che tutti i componenti dell'intero Team Scrum (Team di Sviluppo, Product Owner e Scrum Master), hanno compreso a fondo ogni aspetto che riguardano il lavoro da svolgere.

Alla riunione possono partecipare, se invitati, anche tecnici e altri professionisti per fornire una consulenza più accurata sulla fattibilità pratica di quanto è determinato nel meeting. Tuttavia non possono e non devono disturbare la riunione. Sarà eventualmente compito dello Scrum Master “difendere” da ciò il Team di Sviluppo.

Qualora ci fossero degli item previsti che però il Team non è riuscito a portare a termine durante uno Sprint, il Product Owner li esclude dalla review e li inserisce nuovamente nella lista del Product Backlog. Questo perché durante uno Sprint non è possibile modificarne gli obiettivi e le caratteristiche decise durante la pianificazione dello Sprint Planning. Quindi qualsiasi modifica necessaria e non prevista in cui si incorra, deve essere ritenuta sospesa e presentata poi durante la pianificazione dello Sprint successivo.

Daily Stand-up

OGNI GIORNO DEL CICLO SPRINT SI TIENE IL DAILY STAND-UP PER DETERMINARE E CONDIVIDERE CON IL RESTO DEL GRUPPO, IL LAVORO DA SVOLGERE DURANTE LA GIORNATA

Durante lo svolgimento dello Sprint, quotidianamente ha luogo un altro genere di Evento Scrum: il Daily Stand-up.

Contrariamente allo Sprint Planning, il Daily Stand-up, come del resto indica la parola, è un incontro previsto per ogni giorno del ciclo di Sprint. Questo evento deve avere una durata di quindici minuti al massimo e si tiene, generalmente, sempre alla stessa ora e nello stesso luogo prima che ci si metta a lavorare.

Il Daily Stand-up aumenta il senso di responsabilità del gruppo di lavoro poiché pianifica in anticipo, ciò che dovrà essere svolto e portato a termine durante la giornata, visionando l'operato che già si è prodotto il giorno prima. Il Team di Sviluppo, durante questo meeting, quindi, ispeziona il lavoro che già ha svolto e determina ciò che verrà prodotto nell'immediato e le modalità in cui ogni sviluppatore collaborerà con gli altri al fine di raggiungere l'obiettivo espresso nello Sprint Goal.

Ogni Team di Sviluppo conduce questo evento Scrum nel modo che ritiene più opportuno e adatto al gruppo. Alcuni team utilizzano delle domande base durante l'incontro, altri discutono liberamente. Tuttavia, al di là delle modalità con cui si tiene l'incontro, questo evento è funzionale al Team di Sviluppo poiché solo in questo modo, ogni componente del gruppo può avere ben chiaro ciò che ognuno ha prodotto, come intende lavorare nell'imminente giornata e cosa pensa di riuscire a produrre. Al Daily Stand-up possono partecipare anche altre persone.

Sarà anche in questo caso, compito dello Scrum Master assicurarsi che lo svolgimento di questo meeting proceda senza interruzioni, essendo un incontro interno al Team di Sviluppo e prettamente funzionale solo allo stesso. Lo Scrum Master controllerà anche che si rispetti la durata dell'incontro non superando il tempo previsto dei quindici minuti. Infine ascolta attivamente ciò che viene esposto e apprendere se ci sono eventuali impedimenti che ostacolano la buona riuscita del lavoro, per aiutare il Team di Sviluppo a superarli.

Quando il Daily Stand-up termina, comunque, tutto il gruppo è pronto a lavorare.

Questo evento, quindi, è particolarmente funzionale per instaurare una buona comunicazione all'interno del gruppo ma anche per individuare ed eliminare eventuali ostacoli che intralcerrebbero la buona riuscita del prodotto dello Sprint. In questo modo si evita il rallentamento dell'andamento del lavoro dovuto a errori o altri "intoppi", ispezionando quotidianamente il lavoro svolto e adattandolo all'obiettivo che è stato prefissato.

Sprint Review

L'INFORMALE SPRINT REVIEW PER REVISIONARE IL LAVORO SVOLTO DURANTE IL CICLO SPRINT.

Questo incontro è in realtà un evento informale che si organizza generalmente al termine di un ciclo Sprint.

All'incontro dello Sprint Review partecipa anche lo stakeholder che, insieme a tutto il Team Scrum, verifica i risultati prodotti dall'iterazione.

Come tutti gli eventi Scrum, anche questo meeting possiede una durata limite da rispettare. Il tempo prestabilito per questo incontro è infatti, di un'ora per ogni settimana di durata dello Sprint. Ciò vale a dire che può durare massimo quattro ore se il ciclo di Sprint a cui fa riferimento è di un mese (e quindi di quattro settimane). Proporzionalmente avrà una durata inferiore per un'iterazione più breve. Durante uno Sprint Review,

Il Team Scrum discute con lo stakeholder rispetto all'Incremento di Prodotto realizzato e rilasciato al termine dello Sprint. Il lavoro svolto viene revisionato ispezionando l'Incremento di Prodotto e discutendo su come procedere nel lavoro di sviluppo. In questa occasione il Team di Sviluppo raccoglie i feedback rispetto al lavoro che ha svolto, lo stakeholder ha l'opportunità di esprimere le proprie opinioni o richieste relative al prossimo Sprint, mentre il Product Owner aggiorna la lista del Product Backlog sulla base delle decisioni che emergono durante questo meeting. Lo Sprint Review quindi fornisce una panoramica completa sull'Incremento di Prodotto completato durante lo Sprint, dando un prezioso contributo al successivo Sprint Planning e creando, nell'immediato, un'ulteriore occasione per focalizzarsi sull'obiettivo da raggiungere rispetto al lavoro che si è già portato a termine.

Retrospective Meeting

IL RETROSPECTIVE MEETING ISPEZIONA NEL DETTAGLIO COME È STATO REALIZZATO IL PRODOTTO DEL CICLO SPRINT (STRUMENTI, INTERAZIONI ALL'INTERNO DEL GRUPPO ED EVENTUALI OSTACOLI) AL FINE DI MIGLIORARNE LE MODALITÀ DI LAVORAZIONE

Il Retrospective Meeting è l'ultimo evento che si tiene durante un processo Scrum. Questo incontro viene organizzato generalmente subito dopo lo Sprint Review. Anche questo meeting è time-boxed e come lo Sprint Review, che lo precede ha un tempo limite prestabilito di un'ora per ogni settimana del ciclo Sprint. Tuttavia, a differenza del precedente, si tratta di un evento formale.

Durante il Retrospective Meeting, il Team setaccia al dettaglio gli strumenti utilizzati durante lo Sprint, come i componenti del team si sono relazionati tra loro e tutti i dettagli che riguardano il lavoro stesso. In pratica, come è evidenziato dal nome di questo evento, il Team Scrum analizza in retrospettiva, lo Sprint appena concluso.

Vengono così identificati gli elementi che costituiscono il potenziale di miglioramento del Team per creare un piano da attuare proprio per migliorare ulteriormente il lavoro durante il prossimo Sprint.

Ogni Team Scrum conduce questo incontro nel modo che preferisce.

Tuttavia esiste una consuetudine comune che è quella di discutere segnando su una lavagna una tabella.

Qui si tracciano tre colonne corrispondenti a tre precise determinazioni: cosa è andato bene, cosa deve essere migliorato e dei punti su cui si decide di discutere in seguito, chiamati "action point". Successivamente ad ogni membro del Team vengono distribuiti dei post-it su cui ognuno sintetizza con una frase le proprie osservazioni in maniera libera e il più obiettivamente possibile. Non è consentito, infatti, usare questa occasione per lanciare attacchi personali e polemizzare inutilmente sul lavoro degli altri.

A questo punto lo Scrum Master chiama alla lavagna un membro alla volta che ha così la possibilità di spiegare brevemente il suo pensiero ed attaccare il post-it nella colonna corrispondente. Quando tutti i partecipanti avranno espresso le loro idee, lo Scrum Master raggrupperà i post-it per tematiche. Si passa quindi alla fase delle votazioni.

Ogni componente del Team ha a sua disposizione, un numero di voti corrispondente alla terza parte del numero totale dei temi identificati attraverso i post-it, più uno (assimilabile alla formula matematica $(n/3)+1$). Attraverso la votazione viene assegnato ad ogni tematica un numero di voti che ne identificano la priorità. I temi che presentano il numero maggiore, diventano oggetto di discussione, mentre saranno archiviati quelli che non hanno ottenuto voti.

Il Retrospective meeting deve essere considerata come un'opportunità costruttiva in cui Team Scrum cerca di migliorare il processo e la relativa esecuzione.

In questo senso, quindi, il Retrospective Sprint è un'occasione formale per ispezionare il processo di lavorazione proprio e interno al Team Scrum per migliorare le modalità dell'operato, fermo restando le fasi e i valori previsti dal processo Scrum.

Meeting “aggiuntivi”:

Scrum of Scrums e Product Backlog Refinement

- Lo scrum of scrum si tiene qualora il lavoro sia svolto da più di un team scrum, per coordinarne la lavorazione.
- Il product backlog refinement serve a ridefinire di volta in volta il product backlog. Esso può essere organizzato:
 - *Più volte per cicli sprint lunghi*
 - *Almeno una volta per sprint brevi, prima dello sprint*
 - *Planning del prossimo sprint*

Quando il lavoro è svolto da più di un Team Scrum oppure per attività di sviluppo più complesse, è possibile organizzare altre due tipologie di incontri: lo Scrum of Scrum e il Product Backlog Refinement. Entrambi, comunque, sono incontri funzionali al coordinamento del lavoro di sviluppo.

Lo Scrum of Scrums è un incontro che ha luogo generalmente dopo il quotidiano Daily Stand-up, con cadenza giornaliera oppure saltuariamente. Questo evento serve a coordinare il lavoro tra più Team Scrum creando un'occasione per discutere dell'operato ed evitare sovrapposizioni, integrando al meglio il lavoro svolto tra i vari gruppi.

Lo Scrum of Scrums viene condotto generalmente nelle stesse modalità del Daily Stand-up con poche differenze rispetto al meeting quotidiano. Innanzitutto partecipa una sola persona designata per ogni Team Scrum che svolge il lavoro. In secondo luogo, durante questo incontro viene analizzato il lavoro svolto da ogni Team Scrum cercando di definire quanto è più possibile ciò che si porterà a termine in un futuro imminente, il lavoro che già si è prodotto e se è possibile una maggiore collaborazione tra i team qualora una delle squadre avesse prodotto qualcosa che potesse esser utilizzato dagli altri team. Insomma questo evento è più che mai funzionale a coordinare il lavoro svolto tra più Team Scrum.

Il Product Backlog Refinement, invece, è un incontro che è consigliabile tenere almeno una volta durante l'intero ciclo dello Sprint, o comunque prima di organizzare il prossimo Sprint Planning.

Questa attività è conosciuta anche con il nome di Backlog Grooming, poiché attraverso questo breve incontro si intende “rifinire” o “ripulire” (il termine grooming in italiano significa proprio “governare/ripulire”), i requisiti che deve avere il prodotto da realizzare e che sono descritti nel Product Backlog. In un aggiornamento del 2013 è stato però rimosso ufficialmente il termine “grooming” dalla guida di Scrum poiché questo termine può essere frainteso. È stato ritenuto preferibile, a tal proposito, utilizzare la parola “refinement” e quindi il Backlog Grooming è da allora ufficialmente chiamato: Product Backlog Refinement.

Si tratta di un processo secondo cui si analizza e si stima il relativo Sprint Backlog organizzato per l'imminente Sprint, per ottimizzare il lavoro da svolgere durante l'iterazione. Ciò è reso necessario dal fatto che nella lista del Product Backlog e conseguenzialmente nello Sprint Backlog che ne deriva per ogni ciclo Sprint, si identificano semplicemente i requisiti, le funzionalità che il prodotto da rilasciare deve possedere. Tuttavia non vengono chiariti i dettagli necessari per sviluppare le funzionalità e ciò può creare dubbi ed incertezze di cui bisogna discutere.

È in questo senso che l'evento serve quindi a raffinare ed aggiornare di volta in volta, il Product Backlog aggiungendo o eliminando dettagli in base alle stime effettuate, in modo da allineare il lavoro svolto.

Quando è necessario organizzare questo incontro durante lo sviluppo dello Sprint, allora il lavoro fa riferimento alle singole user story (o storie) dello Sprint. Con il termine user story, si definiscono le varie "caratteristiche" di un software. Organizzare il lavoro di sviluppo di un software in storie, significa suddividere il lavoro in unità più piccole a cui dedicarsi in un tempo limitato. Esistono due tipologie di storie: le Epics sono storie più grandi e quindi di maggiore portata lavorativa; e le user story che sono invece più piccole (contando su un numero ridotto di attività da svolgere).

Attraverso la stima del lavoro svolto, è possibile ispezionare e verificare le storie completate o da svolgere.

Il Backlog Grooming serve a focalizzarsi maggiormente sull'obiettivo da portare a termine e ispeziona il lavoro svolto per eliminare le storie completate o che risultano ormai "fuori tema" rispetto all'obiettivo da raggiungere. Questo incontro, però, più che durante il periodo dell'iterazione, è generalmente organizzato sul finire della stessa e quindi prima del prossimo ciclo Sprint. Precede, quindi, lo Sprint Planning dell'iterazione successiva. In questo caso il processo di stima non fa più riferimento alle singole storie e quindi allo Sprint Backlog, ma fa riferimento ad una concezione del lavoro svolto secondo una visuale più generica e complessiva. Si analizza quindi ciò che si è svolto e che risulta dal ciclo dello Sprint e quindi il discorso si rapporta spostandosi maggiormente sul Product Backlog più che sullo Sprint Backlog.

Resta comunque responsabilità del Product Owner, rendere chiare a tutto il Team Scrum, le priorità, sia che queste facciano riferimento al lavoro quotidiano, o che facciano parte di un discorso più generale.

Per Sprint di durata maggiore è consigliabile, organizzare più volte questo incontro. Viceversa se la durata del ciclo dell'iterazione è breve, l'incontro può limitarsi anche ad una sola volta, comunque entro il termine dello Sprint. Il metodo più comune per stimare il lavoro svolto o ancora da svolgere, è quello chiamato "planning poker". Tale metodo prevede una discussione tra tutti i membri del team giustificando e valutando le stime che ognuno ha effettuato affinché si giunga ad una stima condivisa e concorde a tutti.

Durante questo metodo si assegnano alle varie user story, degli story point, cioè si puntualizzano i vari elementi da svolgere ordinandoli in base alle loro priorità.

Questa attività, sebbene sia spesso trascurata durante il processo Scrum, è in realtà molto importante poiché è un'ulteriore occasione poiché rende trasparente il lavoro svolto o da svolgere nel prossimo Sprint e attraverso questa attività si ispeziona l'operato fornendo a tutti l'opportunità di adattare al meglio il lavoro.

Definition of Done

LA DEFINITION OF DONE È UNA CONVENZIONE CON CUI SI STABILISCONO LE CARATTERISTICHE CHE IL PRODOTTO DEVE AVERE PER POTER ESSERE DEFINITO COMPLETATO (“FATTO”/“FINITO”).

La Definition of Done (abbreviata come DoD) è tradotta in italiano letteralmente come Definizione di Fatto (o Finito). Essa è una convenzione prestabilita a monte dal Team Scrum, prima che ci si adoperi effettivamente nella lavorazione relativa al ciclo Sprint.

Generalmente i parametri della Definition of Done vengono prestabiliti e messi in chiaro durante lo Sprint Planning, affinché tutti i componenti del Team la comprendano a pieno, senza riservare alcun dubbio in merito. Tale convenzione si applica quindi al lavoro da completare.

Alla fine di ogni iterazione, bisogna realizzare un artefatto denominato Incremento di Prodotto che rappresenta il risultato della lavorazione condotta durante il periodo dello Sprint. Le caratteristiche e funzionalità di questo prodotto devono poter essere sommate a quelle degli altri incrementi rilasciati dai precedenti Sprint, in modo che il software che deve essere rilasciato (e che quindi si sta sviluppando), risulti un prodotto “completo” e funzionante in ogni sua parte.

Tuttavia ognuno di noi ha un'ideologia di “fatto” o “completato” chiaramente diversa dall'altro. Così anche all'interno di un Team di sviluppatori, per alcuni si può considerare fatto un incremento che sia ben documentabile, per altri se questo è stato o può essere codificato, per altri ancora che rispetti la propria personale idea di progettazione iniziale e via dicendo. Per questo motivo è necessario eliminare qualsiasi possibile fraintendimento, definendo una checklist completa di tutte le caratteristiche che un prodotto deve possedere per poter essere comunemente dichiarato “fatto”, quale è appunto la DoD.

In ultima analisi bisogna puntualizzare che, data la forte accezione di lavoro implementare ed empirico del processo Scrum, la Definition of Done di un Incremento di Prodotto non si riferisce solo al singolo prodotto immediatamente realizzato.

Ogni nuovo Incremento di Prodotto realizzato, infatti, può considerarsi completato (e quindi in linea con la DoD) solo quando viene completamente testato e risulta contenere al suo interno le funzionalità di tutti i prodotti rilasciati dai precedenti Sprint. Quando ciò accade, l'Incremento di Prodotto risulta "funzionante" e quindi può essere definito da tutto il Team con l'accezione di "fatto".

Soltanto in questo modo si può garantire che tutti gli elementi richiesti nel Product Backlog, che siano già stati completati in precedenza, continuino a funzionare regolarmente anche se implementati dalle ulteriori funzionalità dell'ultimo Incremento di Prodotto.