



Una Guida Snella alla Teoria ed alla Pratica di Scrum Versione 2.0

Traduzione italiana di Salvatore Reale www.gpnitalia.it

Pete Deemer
GoodAgile
www.goodagile.com

Gabrielle Benefield
Evolve
www.evolvebeyond.com

Craig Larman
www.craiglarman.com

Bas Vodde
Odd-e
www.odd-e.com



Una nota per i lettori: Ci sono molte descrizioni sintetiche di Scrum disponibili online, e questo brevissimo trattato mira a fornire il successivo livello di dettaglio sulle pratiche di Scrum. Non va inteso come atto finale di una formazione Scrum, team che stanno prendendo in considerazione l'adozione di Scrum sono invitati a dotarsi di *Agile Project Management with Scrum* di Ken Schwaber, oppure *Succeeding with Agile* di Mike Cohn e cogliere i vantaggi dei tanti corsi di formazione Scrum e delle ottime opzioni di coaching che sono disponibili; tutte le informazioni sono su scrumalliance.org. I nostri ringraziamenti vanno a Ken Schwaber, Dr. Jeff Sutherland e Mike Cohn per il loro generoso contributo.

L'ultima versione del Primer si trova su: http://www.infoq.com/minibooks/Scrum_Primer

Le traduzioni si trovano su: <http://www.scrumprimer.org/>

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

Al di là dello sviluppo tradizionale

Lo sviluppo tradizionale fondato su gruppi mono-disciplinari, cicli di feedback deboli o tardivi, una pianificazione predittiva anticipata e un flusso sequenziale dall'analisi al test non è molto efficace in un mondo che ha nella volatilità la sua caratteristica più attuale. Questo approccio ritarda le risposte, l'apprendimento, e il potenziale ritorno sugli investimenti a causa dell'assenza di software realmente funzionante fino a quando il gioco è alla fine, causando la mancanza di trasparenza, la mancanza di capacità di migliorare, la riduzione della flessibilità, e un aumento dei rischi aziendali e tecnici.

Un'alternativa – basata su team cross-funzionali e sullo sviluppo iterativo - esisteva da decenni, senza essersi largamente diffusa quanto il modello tradizionale.

La metodologia Scrum ha dimostrato in un contesto semplice alcuni concetti per lo sviluppo di un prodotto, tra cui: team reali, team cross-funzionali, team auto-gestiti, brevi cicli di feedback iterativi e completi, abbassamento del costo del cambiamento. Questi concetti aumentano l'agilità e il feedback, consentono di anticipare il ROI e di ridurre i rischi.

Panoramica

Scrum è uno schema di sviluppo in cui i team cross-funzionali sviluppano prodotti o progetti in modo iterativo, incrementale. Lo sviluppo è strutturato in cicli di lavoro chiamati **Sprint**. Queste iterazioni non durano più di quattro settimane ciascuno (più comunemente due settimane), e avvengono una dopo l'altra, senza pause. Gli Sprint sono a tempo limitato (*timeboxed*) - finiscono ad una data fissata, che il lavoro sia stato completato o meno, e non sono *mai prolungati*. Di solito i Team Scrum scelgono una durata dello Sprint e la mantengono per tutti gli Sprint successivi fino a quando essi non migliorano al punto da poter utilizzare un ciclo più breve. All'inizio di ogni Sprint, un **Team cross-funzionale** (di circa sette persone) seleziona **le voci** (items, requisiti del cliente) da un elenco di priorità. Il Team concorda su un obiettivo collettivo riguardo a ciò che ritiene di poter consegnare entro la fine dello Sprint, qualcosa che sia tangibile e sarà veramente "fatto". Durante lo Sprint, non possono essere aggiunti nuovi contenuti; Scrum accetta cambiamenti per il *prossimo* Sprint, ma il breve Sprint attuale ha lo scopo di concentrarsi su un obiettivo piccolo, chiaro, relativamente stabile. Ogni giorno il Team si riunisce brevemente per controllare il suo progresso, e identificare i prossimi passi necessari per completare il lavoro rimanente. Alla fine dello Sprint, il Team revisiona lo Sprint con i soggetti interessati, e dimostra ciò che ha costruito. Le persone ottengono riscontri che possono essere integrati nel prossimo Sprint. Scrum pone l'enfasi sul prodotto funzionante alla fine dello Sprint e che sia realmente completato; nel caso del software, questo significa un sistema che è integrato, completamente testato, documentato per l'utente finale, e potenzialmente rilasciabile. I ruoli chiave, gli strumenti e gli eventi sono riassunti nella Figura 1.

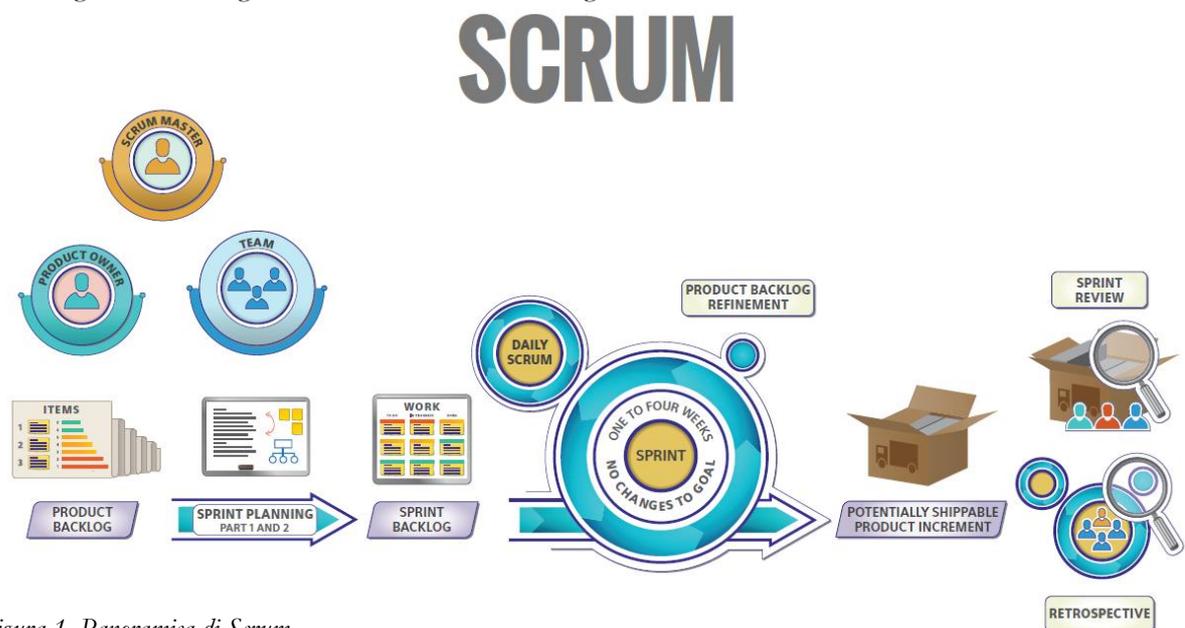


Figura 1. Panoramica di Scrum

Un tema importante in Scrum è "ispeziona e adegua." Dal momento che lo sviluppo comporta inevitabilmente apprendimento, innovazione e sorprese, Scrum insiste sul prendere in esame una breve fase di sviluppo, ispezionare sia il prodotto risultante sia l'efficacia delle pratiche correnti, e poi adeguare gli obiettivi di prodotto e le pratiche di processo. *Ripetendo indefinitamente.*

Ruoli in Scrum

In Scrum ci sono tre ruoli: Product Owner, Team, e Scrum Master. Insieme, sono conosciuti come il Team Scrum.

Il **Product Owner** ha la responsabilità di massimizzare il ritorno sugli investimenti (ROI), di identificare le caratteristiche del prodotto, da tradursi in una lista di priorità, di decidere cosa dovrebbe andare in cima alla lista per il prossimo Sprint, e di riassegnare le priorità e perfezionare l'elenco con continuità. Il Product Owner ha la responsabilità economica per il prodotto, ammesso che sia un prodotto commerciale. Nel caso si tratti di una applicazione interna, il Product Owner non è responsabile per il ROI come si intende per un prodotto commerciale (che genererà entrate), ma è ancora responsabile di massimizzare la scelta - ad ogni sprint - delle caratteristiche di prodotto che portano il più alto valore. In pratica, 'valore' è un termine sfocato, e l'assegnazione di una priorità può essere influenzata dal desiderio di soddisfare i clienti chiave, dall'allineamento con gli obiettivi strategici, dalla necessità di affrontare i rischi, di migliorare, e anche da altri fattori. In alcuni casi, il Product Owner e il cliente sono la stessa persona, questo avviene comunemente per le applicazioni interne. In altri, il cliente potrebbe essere costituito da milioni di persone con una varietà di esigenze diverse, nel qual caso il ruolo di Product Owner è simile alle posizioni di Product Manager e Product Marketing Manager di molte organizzazioni di prodotto. Tuttavia, il Product Owner è un ruolo un po' diverso rispetto a un tradizionale Product Manager perché il primo interagisce attivamente e regolarmente con il Team, assegna le priorità lavorando con tutte le parti interessate e rivedendo i risultati ad ogni Sprint, piuttosto che delegare le decisioni sullo sviluppo ad un project manager. E' importante osservare che in Scrum vi è una ed una sola persona che agisce come - ed ha l'autorità finale di - un Product Owner, e solo lui o lei è responsabile per il valore del lavoro, sebbene questa persona non debba lavorare da sola.

Il Team (chiamato anche Team di Sviluppo) costruisce il prodotto che il Product Owner indica: l'applicazione o il sito Web, per esempio. Il Team Scrum è "cross-funzionale", vale a dire che include tutte le competenze necessarie per rilasciare potenzialmente il prodotto ad ogni Sprint, ed è "auto-organizzato" (auto-gestito), con un elevato grado di autonomia e responsabilità. Il Team decide quante voci (dal set offerto dal Product Owner) realizzare in uno Sprint, e il modo migliore per raggiungere tale obiettivo.

Ogni membro di un Team Scrum è solo un *componente del Team*. Si noti che non ci sono titoli specialistici fissati in un gruppo che adotta Scrum, non c'è analista di business, nessun DBA, nessun architetto, nessun capo del Team, nessun interaction/UX designer, nessun programmatore. Essi lavorano insieme durante ogni Sprint in qualunque modo sia appropriato per raggiungere l'obiettivo che si sono auto-assegnati.

Poiché ci sono solo *membri del Team*, il Team non è solo cross-funzionale, ma dimostra anche capacità di *multi-apprendimento*: ogni persona ha certamente punti di forza specifici, ma continua anche ad imparare altre specialità. Ogni persona avrà competenze di primo, secondo e anche terzo livello, ci si aspetta che "vada dove c'è del lavoro da fare", e che individualmente si assumano compiti anche in aree meno familiari, se ciò aiuta a completare un'attività. Ad esempio, una persona la cui principale abilità è il design di interazione potrebbe avere una competenza secondaria in test automatizzati; qualcuno con l'abilità primaria in scrittura tecnica potrebbe anche aiutare con l'analisi e la programmazione.

Il Team di Scrum è di sette più o meno due persone, e per un prodotto software il Team potrebbe includere persone con competenze in analisi, sviluppo, test, progettazione di interfacce, progettazione di database, architettura, documentazione, e così via.

Il Team sviluppa il prodotto e fornisce idee al Product Owner su come rendere il prodotto migliore. In Scrum i Team sono più produttivi ed efficaci se tutti i membri sono al 100 per cento dedicati al lavoro per un unico prodotto durante lo Sprint, il Team evita il multi-tasking su più prodotti o progetti, al fine di evitare il costo di dividere l'attenzione e del frequente cambiamento di contesto.

Team stabili sono associati a una maggiore produttività, quindi evitate di cambiare i membri del Team. Gruppi di prodotto con molte persone sono organizzati in più Team, ciascuno focalizzato su diverse funzionalità del prodotto, con uno stretto coordinamento dei loro sforzi. Dal momento che un Team spesso fa tutto il lavoro (pianificazione, analisi, programmazione e test) per una funzionalità centrata sul cliente completa, i Team sono noti anche come *feature Teams*.

Lo **ScrumMaster** aiuta il gruppo di prodotto ad apprendere e applicare Scrum per conseguire valore di business. Lo ScrumMaster fa tutto ciò che è in suo potere per aiutare il Team, il Product Owner e l'organizzazione ad avere successo. Lo ScrumMaster *non* è il manager dei membri del Team, né è un project manager, team leader, o rappresentante del team. Invece, lo ScrumMaster *serve* il Team, aiuta a rimuovere gli ostacoli, protegge il Team da interferenze esterne, e aiuta il Team ad adottare moderne pratiche di sviluppo. Lo ScrumMaster educa, allena e guida il Product Owner, il Team e il resto dell'organizzazione nell'uso sapiente di Scrum. Lo ScrumMaster è un *allenatore* e un *insegnante*. Lo ScrumMaster fa in modo che tutti (compreso il Product Owner, e coloro che sono nel management) capiscano i principi e le pratiche di Scrum, e aiuta a guidare l'organizzazione attraverso il cambiamento spesso difficile ma necessario per raggiungere il successo con lo sviluppo agile.

Poiché Scrum rende visibili molti ostacoli e le minacce all'efficacia del Team e del Product Owner, è importante avere uno ScrumMaster impegnato a lavorare energicamente per aiutare a risolvere questi problemi, o per il Team e per il Product Owner sarà difficile avere successo. Ci dovrebbe essere uno ScrumMaster dedicato e a tempo pieno, anche se in un Team più piccolo si potrebbe avere un membro del Team che ricopre questo ruolo (quando lo fa avrà un carico di lavoro implementativo più leggero). Bravi ScrumMasters possono provenire da qualsiasi precedente incarico o disciplina: Ingegneria, Progettazione, Test, Product Management, Project Management, o Gestione della Qualità.

Lo ScrumMaster e il Product Owner non possono essere lo stesso individuo in quanto i loro obiettivi sono così diversi che la loro combinazione spesso porta a confusione e conflitto. Un comune e spiacevole risultato della combinazione di questi ruoli è un Product Owner micro-manager, che è in contraddizione con i Team autogestiti che Scrum richiede. A differenza di un manager tradizionale, lo ScrumMaster non dice alle persone cosa fare né assegna compiti - facilita il processo, sostenendo il Team quando si organizza e si auto-gestisce. Se lo ScrumMaster in precedenza era in una posizione di manager per il Team, dovrà cambiare in modo significativo la sua mentalità e lo stile di interazione con il Team per avere successo con Scrum.

Nota: non c'è affatto un ruolo di project manager in Scrum. Questo perché non è necessario; le responsabilità tradizionali di un project manager sono state suddivise e riassegnate tra i tre ruoli di Scrum, e soprattutto al Team e al Product Owner, piuttosto che allo ScrumMaster. Praticare Scrum con l'aggiunta di un project manager indica un fraintendimento fondamentale di Scrum, e in genere si traduce in responsabilità in conflitto, autorità poco chiara, e risultati non ottimali.

A volte un (ex) project manager può passare nel ruolo di ScrumMaster, ma il successo di questo approccio è fortemente dipendente dall'individuo, e da come abbia ben compreso la differenza fondamentale tra i due ruoli, sia nella responsabilità quotidiana e sia nella mentalità necessaria per avere successo. Un buon modo per capire a fondo il ruolo dello ScrumMaster e iniziare a sviluppare le competenze di base necessarie per riuscire nel ruolo, è quello di partecipare ad un corso Scrum Alliance per ScrumMaster Certificato.

Oltre a questi tre ruoli, ci sono altri soggetti che contribuiscono al successo del prodotto, compresi i dirigenti, i clienti e gli utenti finali. Alcuni soggetti interessati, quali i manager funzionali (ad esempio, un responsabile tecnico), possono trovare che il loro ruolo, pur prezioso, cambia quando si adotta Scrum. Per esempio:

- essi sostengono il Team rispettando le regole e lo spirito di Scrum
- contribuiscono a rimuovere gli ostacoli identificati dal Team e dal Product Owner
- mettono a disposizione la loro competenza ed esperienza

In Scrum, questi individui sostituiscono il tempo che trascorrevano a svolgere il ruolo di "tate" (assegnazione di compiti, ottenere rapporti di stato, e altre forme di microgestione) con tempo in

cui svolgono il ruolo di "guru" e "al servizio" del Team (mentoring, coaching, contribuire a rimuovere gli ostacoli, aiutare a risolvere problemi, fornire input creativo, e guidare lo sviluppo delle competenze dei membri del Team). In questo cambiamento, i manager possono avere bisogno di cambiare il loro stile di gestione, ad esempio, instillando dubbi socratici per aiutare il Team a scoprire la soluzione di un problema, piuttosto che semplicemente decidere una soluzione e assegnarla al Team.

Product Backlog

Quando un gruppo ha in programma una transizione a Scrum, prima che il primo Sprint possa iniziare, hanno bisogno di un **Product Backlog**, un elenco di funzionalità centrate sul cliente in ordine di priorità (1, 2, 3, ...).

Il Product Backlog esiste (e si evolve) per tutta la durata del prodotto, è la tabella di marcia del prodotto (**Figura 2 e Figura 3**). In qualsiasi momento, il Product Backlog è l'unica vista definitiva su "tutto ciò che potrebbe essere fatto dal Team d'ora in avanti, in ordine di priorità". Esiste un solo Product Backlog per un prodotto, questo significa che il Product Owner è tenuto a prendere decisioni per assegnare le priorità a tutto spettro, rappresentando gli interessi di tutti i soggetti interessati (incluso il Team).

Priorità	Voce	Dettagli (wiki URL)	Stima iniziale dimensioni	Nuove stime allo Sprint ...					
				1	2	3	4	5	6
1	Come acquirente, voglio inserire un libro nel carrello d'acquisto (vedi esempi UI sulla wiki page)	...	5						
2	Come acquirente, Voglio rimuovere un libro dal carrello d'acquisto	...	2						
3	Migliorare le prestazioni del processo di transazione (vedi le metriche sui target di prestazioni sul wiki)	...	13						
4	Investigare soluzioni per velocizzare la validazione della carta di credito	...	20						
5	Aggiornare tutti i server ad Apache 2.2.3	...	13						
6	Eseguire diagnosi e correggere gli errori dello script di elaborazione dell'ordine (bugzilla ID 14823)	...	3						
7	Come acquirente voglio creare e salvare una lista dei desideri	...	40						
8	Come acquirente, voglio aggiungere o cancellare voci nella mia lista dei desideri	...	20						

Figura 2. Il Product Backlog



Figure 3. Visual Management: Voci del Product Backlog a parete

Il Product Backlog include una varietà di **voci**, soprattutto nuove funzionalità per il cliente ("consentire a tutti gli utenti di inserire un libro nel carrello della spesa"), ma anche *i principali* obiettivi di miglioramento dell'ingegneria (ad esempio, "riscrivere il sistema da C++ a Java"), obiettivi di miglioramento (ad esempio, "accelerare i nostri test"), lavori di ricerca ("studiare soluzioni per accelerare la validazione della carta di credito"), e, possibilmente, difetti noti ("diagnosticare e correggere gli errori dello script di elaborazione degli ordini") se ci sono solo pochi problemi. (Un sistema con molti difetti di solito ha un sistema di tracciamento dei difetti dedicato.)

Le voci del Product Backlog sono articolate in qualsiasi modo che sia chiaro e sostenibile. Contrariamente a malintesi comuni, il Product Backlog non contiene "user stories"; contiene semplicemente *voci*. Tali voci possono essere espresse come user stories, casi d'uso, o utilizzando altri approcci per esprimere i requisiti che il gruppo trovi utili. Ma qualunque sia l'approccio, la maggior parte delle voci dovrebbero concentrarsi sul fornire valore ai clienti.

Un buon Product Backlog è DEEP...

Dettagliato in modo appropriato. Le voci di priorità superiore sono molto più affinate e dettagliate rispetto alle voci di priorità più bassa, dal momento che la prima saranno da lavorare subito rispetto alle seconde. Ad esempio, il 10% nella parte superiore del backlog può essere composto da voci molto piccole, ben analizzate, e l'altro 90% molto meno.

Stimato (Estimated). Le voci per la release corrente hanno bisogno di essere stimate, e, inoltre, dovrebbero essere nuovamente stimate ad ogni Sprint in quanto tutti imparano e si acquisiscono nuove informazioni. Il Team fornisce al Product Owner le stime dello *sforzo* per ogni voce del Product Backlog, e forse anche le stime di *rischio tecnico*. Il Product Owner e gli altri soggetti coinvolti nel business forniscono informazioni sul valore dei requisiti di prodotto, che possono includere le entrate acquisite, i costi ridotti, rischi aziendali, l'importanza per i vari soggetti interessati, e altro ancora.

Emergente.

In risposta all'apprendimento e alla variabilità, il Product Backlog viene regolarmente affinato. Ad ogni Sprint, le voci possono essere aggiunte, rimosse, modificate, divise, e cambiate in priorità. Così, il Product Backlog viene continuamente aggiornato dal Product Owner per riflettere i cambiamenti delle esigenze del cliente, nuove idee o intuizioni, le mosse della concorrenza, ostacoli tecnici che si presentano, e così via.

Prioritizzato.

Le voci nella parte superiore del Product Backlog hanno la priorità assegnata o ordinata da 1 a N. In generale, gli elementi a più alta priorità devono fornire il maggior *vantaggio economico*: un'alta redditività (valore di business) a basso costo. Un'altra motivazione per aumentare la priorità di un elemento è quello di *affrontare presto i rischi elevati, prima che questi rischi si manifestino*.

Lo sviluppo tradizionale di solito non pone l'enfasi sul rilascio in base al maggior vantaggio economico, ma questo è un tema di Scrum, e pertanto il Product Owner dovrà imparare a valutare l'impatto del "valore di business." Questo è qualcosa in cui lo ScrumMaster può aiutare il Product Owner ad imparare. Che cosa significa "valore di business"? Alcuni Team di prodotto utilizzano una semplice stima dei "value-points" (punti-di-valore) relativa per ciascun elemento del Product Backlog che sintetizza una "stima approssimativa" di fattori tra cui l'aumento dei ricavi, riduzione dei costi, le preferenze degli stakeholder, la differenziazione del mercato, e così via.

In alcuni casi una voce specifica del backlog viene finanziata attraverso uno o più clienti che pagano per il suo sviluppo e in tal modo si può utilizzare il valore esatto (a breve termine) delle entrate previste per quella voce come stima del valore. In altri casi una stima del valore specifica per voce è troppo sfumata o grossolana, si applica quindi un approccio più ampio basato sui risultati di business ("aumentare gli abbonamenti del 10% entro settembre") in cui il valore viene rilasciato solo quando diversi elementi che contribuiscono al risultato economico vengono rilasciati insieme. In tal caso, il Product Owner deve definire l'incremento successivo del Minimum Viable Product (MVP: si tratta di una versione iniziale del prodotto costruita in modo tale da ricomprendere il set minimo di caratteristiche indispensabili al Team per capire cosa desiderano i clienti).

Per le stime di sforzo, una tecnica comune è quella di stimare in termini di dimensione relativa (fattore di sforzo, complessità, incertezza) utilizzando l'unità "story points" o semplicemente "punti".

Questi sono solo suggerimenti; Scrum non definisce la tecnica per esprimere o assegnare la priorità alle voci del Product Backlog e non stabilisce la tecnica o l'unità di misura della stima.

Una tecnica comune usata in Scrum è quella di monitorare quanto lavoro si completa ad ogni Sprint. Ad esempio una media di 26 punti completati per Sprint. Con queste informazioni si può tentare una previsione della data di rilascio per completare tutte le funzionalità, oppure quante funzionalità possono essere completate entro una data fissata, se la media si mantiene senza variazioni di rilievo. Questa media è chiamata la "velocità" (velocity). La velocità è espressa nelle stesse unità delle stime dimensionali di ogni voce del Product Backlog.

Le voci nel Product Backlog possono variare in modo significativo in termini di dimensioni o di sforzo. Quelle più grandi sono suddivise in elementi più piccoli durante il workshop di affinamento del Product Backlog o lo Sprint Planning Meeting, e le voci più piccole possono considerarsi consolidate. Le voci nel Product Backlog per gli Sprint più prossimi devono essere sufficientemente piccole e dettagliate da poter essere interpretate dal Team, consentendo previsioni formulate nel corso della riunione di Sprint Planning tali da essere significative; questa è chiamata una dimensione "azionabile".

I principali miglioramenti tecnici che necessitano di molto tempo e denaro dovrebbero essere nel Product Backlog, in quanto possono essere una opzione per ulteriore investimento, in ultima analisi, da effettuarsi in base alla scelta del Product Owner, che dovrebbe essere orientata al business. Si noti che in Scrum, il Team è indipendente nello scegliere quante voci dal Product Backlog inserire in uno Sprint, quindi è libero di includere attività minori di miglioramento dell'ingegneria in quanto queste attività possono essere considerate parte del costo normale del business, e necessarie ad uno sviluppatore per svolgere al meglio il proprio lavoro. Detto questo, in ogni Sprint, la *maggior parte* del tempo di un Team di solito dovrebbe essere dedicato all'obiettivo del Product Owner, non a compiti di ingegneria interna.

Uno dei miti che aleggia su Scrum è che ti impedisce di scrivere specifiche dettagliate. In realtà, spetta al Product Owner e al Team decidere la quantità di dettagli necessaria, e questo varierà da una voce di backlog alla successiva, a seconda dell'intuizione del Team, e di altri fattori. Definite ciò che è importante nella minore quantità di spazio necessaria - in altre parole, non descrivete ogni possibile dettaglio di un soggetto, basta chiarire ciò che è necessario perché sia capito, e accrescere questo livello di dettaglio con il dialogo continuo tra il Team, il Product Owner e gli altri soggetti interessati. Le voci di bassa priorità del Product Backlog, che non saranno lavorate per qualche tempo, di solito sono di grandi dimensioni, con requisiti meno affinati. Le voci di alta priorità e maggiormente affinate del Product Backlog, che saranno presto implementate, devono tendere ad avere maggior dettaglio.

Definizione di “Fatto” (Definition of Done)

Ciò che viene prodotto ad ogni Sprint è ufficialmente chiamato Incremento di Prodotto Potenzialmente Rilasciabile. Prima di iniziare il primo Sprint, il Product Owner, il Team, e lo ScrumMaster devono passare in rassegna tutto ciò che è necessario affinché una voce del Product Backlog si possa considerare potenzialmente rilasciabile. Tutte le attività che sono necessarie per poter rilasciare il prodotto dovrebbero essere incluse nella definizione di Potenzialmente Rilasciabile e, pertanto, dovrebbero essere completate durante lo Sprint.

Purtroppo, quando i Team iniziano ad utilizzare Scrum, spesso non sono in grado di raggiungere l'obiettivo di fornire un incremento potenzialmente rilasciabile ad ogni Sprint. Questo spesso avviene perché il Team è debole nell'automazione o non è sufficientemente cross-funzionale (ad esempio, i redattori tecnici non sono ancora inclusi nel Team cross-funzionale). Nel corso del tempo, il Team deve migliorare in modo tale da essere in grado di fornire un incremento potenzialmente rilasciabile ad ogni Sprint, ma per iniziare, avranno bisogno di stabilire le basi delle loro capacità attuali. Queste sono registrate nella Definizione di Fatto.

Prima del primo Sprint, il Product Owner e il Team hanno bisogno di concordare una Definizione di Fatto, che è un sottoinsieme delle attività che sono necessarie per la creazione di un incremento potenzialmente rilasciabile (per un buon Team, sarà l'intero insieme). Il Team pianificherà il suo lavoro durante lo Sprint in modo concorde a questa Definizione di Fatto.

Un buon Product Owner vorrà sempre che la Definizione di Fatto sia la più vicina possibile al Potenzialmente Rilasciabile in quanto ciò aumenterà la trasparenza nello sviluppo e ridurrà *i ritardi e i*

rischi. Se la Definizione di Fatto non è uguale a Potenzialmente Rilasciabile, allora il lavoro viene ritardato tornando a prima del rilascio che provoca questo *rischio e ritardo*. Questo lavoro in ritardo è talvolta chiamato il *lavoro incompiuto*.

Un Team Scrum dovrebbe migliorare continuamente, ciò si riflette nella evoluzione della propria Definizione di Fatto.

Sprint Planning

Riassunto: Un incontro per preparare lo Sprint, generalmente diviso in due parti (la prima parte sul "cosa" e la seconda parte sul "come").

Partecipanti: Parte Prima: Product Owner, Team, ScrumMaster. Parte Seconda: Team, ScrumMaster, Product Owner (facoltativo, ma dovrebbe rimanere disponibile per eventuali domande)

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

All'inizio di ogni Sprint, si svolge lo **Sprint Planning Meeting**. E' diviso in due incontri distinti, il primo dei quali si chiama **Sprint Planning Parte Prima**.

Nello **Sprint Planning Parte Prima**, il Product Owner e il Team passano in rassegna le voci ad alta priorità nel Product Backlog per il quale il Product Owner è interessato all'implementazione nello Sprint che sta per iniziare. Di solito, queste funzionalità saranno state ben analizzate in un precedente Sprint (durante l'affinamento del Product Backlog), in modo che in questa riunione ci siano solo da chiarire questioni di minor peso dell'ultima ora. In questa riunione, il Product Owner e il Team discutono gli obiettivi e il contesto di questi elementi ad alta priorità nel Product Backlog, fornendo al Team le considerazioni del Product Owner. La prima parte si concentra sulla comprensione di *quali* funzionalità il Product Owner richiede e *perché* sono necessarie. Alla fine della prima parte il Product Owner (che è notoriamente molto impegnato) può lasciare il meeting anche se **deve** rimanere a disposizione (per esempio, per telefono) durante la seconda parte della riunione.

Nella prima parte, il Team e il Product Owner possono anche elaborare lo **Sprint Goal**. Si tratta di una dichiarazione di sintesi dell'obiettivo dello Sprint, che abbia idealmente un tema coesivo. L'obiettivo di Sprint dà anche al Team un ambito di flessibilità per quanto riguarda ciò che essi possono effettivamente consegnare, perché anche se dovesse diventare necessario rimuovere alcune voci (in quanto lo Sprint è timeboxed), essi dovrebbero tuttavia impegnarsi a fornire qualcosa di tangibile e "fatto", che sia nello spirito dello Sprint Goal.

Quanto grandi dovrebbero essere le voci che vengono inserite in uno Sprint? Ogni elemento deve essere diviso in dimensioni sufficientemente piccole in modo che si stimi possa richiedere considerevolmente meno tempo dell'intera durata dello Sprint. Una linea guida comune è che un elemento sia stimato dall'intero Team abbastanza piccolo da potersi completare entro un quarto o meno della durata di uno Sprint.

Lo **Sprint Planning Parte Seconda** si concentra su *come* implementare le voci che il Team si è preso in carico. Il Team prevede la quantità di voci che ritiene di poter completare entro la fine dello Sprint, a partire dalla cima del Product Backlog (in altre parole, a partire dalle voci che sono di massima priorità per il Product Owner) ed elaborando verso il basso la lista nell'ordine. *Questa è una pratica fondamentale in Scrum: è il Team a decidere quanto lavoro porterà a termine, piuttosto che farselo assegnare dal Product Owner*. Questo porta ad una previsione più attendibile perché il Team la effettua sulla base di una propria analisi e pianificazione. Sebbene il Product Owner non abbia il controllo su quanto il Team sottoscrive, sa che le voci sono tratte dalla cima del Product Backlog - in altre parole, le voci che ha valutato come più importanti. Il Team ha la possibilità di fare pressioni per voci che si trovano più in basso nella lista, questo di solito accade quando il Team e il Product Owner si rendono conto che qualcosa di priorità inferiore si adatta facilmente e opportunamente con le voci ad alta priorità.

Lo Sprint Planning Meeting dura spesso diverse ore, ma non più di quattro ore per uno Sprint di due settimane - il Team sta facendo una seria previsione su come completare il lavoro, e questo richiede un'attenta riflessione per avere successo. Parte Prima e Parte Seconda sono della medesima durata limitata, per uno Sprint di due settimane, ciascuna parte è di due ore al massimo.

Scrum non definisce come eseguire esattamente lo Sprint Planning nella seconda parte. Alcuni Team utilizzano la velocità mostrata negli Sprint precedenti come riferimento su quante attività allocarsi. Altri

Team useranno un approccio meno grossolano, calcolando prima la loro capacità.

Quando si utilizza l'approccio per capacità, il Team, nella seconda parte dello Sprint Planning, calcola quanto tempo ogni membro del Team può allocare per il lavoro riconducibile allo Sprint. La maggior parte dei Team assumono che i membri del Team possano concentrarsi sul lavoro dello Sprint solo per 4-6 ore al giorno - il resto del tempo va a e-mail, pausa pranzo, facebook, riunioni e per bere il caffè.

Una volta che la capacità è determinata, il Team ha bisogno di capire quanti elementi di backlog sono in grado di completare nel tempo disponibile, e come procederanno al loro completamento. Questo spesso inizia con una discussione di design davanti ad una lavagna. Una volta che il disegno complessivo è capito, il Team scompone ogni voce di backlog in compiti di lavoro più capillari.

Prima di prendere in considerazione le voci di backlog, il Team può concentrarsi sull'individuare i compiti per un obiettivo di miglioramento creato nella Retrospettiva dello Sprint precedente. Poi, il Team seleziona il primo punto nell'ordine del Product Backlog - massima voce prioritaria del Product Owner - e prosegue verso il basso fino a quando il Team è 'pieno' (totalmente allocato). Per ogni voce i membri del Team creano un elenco dei lavori che consiste nelle voci del backlog decomposti in singoli compiti o, quando le voci del backlog sono già così piccole da richiedere solo un paio d'ore per essere realizzate, le voci sono prese così come sono. Questa lista di lavori da svolgere durante lo Sprint è chiamato **Sprint Backlog** (Figura 4 e Figura 5).

		Nuove stime dello sforzo Rimanente alla fine del giorno...							
Voce del Product Backlog	Sprint Task	Volontario	Stima iniziale dello sforzo	1	2	3	4	5	6
Come acquirente, voglio inserire un libro nel carrello d'acquisto	Modificare il database		5						
	Creare la pagina web (UI)		8						
	Creare la pagina web (Javascript logic)		13						
	Scrivere i test di accettazione automatici		13						
	Aggiornare la pagina di aiuto per l'acquirente		3						
	...								
Migliorare le prestazioni della elaborazione delle transazioni	Inserire il codice DCP e completare i test al livello-layer		5						
	completare la procedura ordini per pRank		8						
	adattare il DCP e il reader a pRank http API		13						

Figura 4. Esempio di un modo per creare uno Sprint Backlog

Alla fine dello Sprint Planning Meeting, il Team fissa un obiettivo realistico che consiste in quello che credono di poter consegnare entro la fine dello Sprint. Tradizionalmente, questo è stato chiamato Sprint Commitment - il Team si impegna a fare del suo meglio per raggiungere l'obiettivo. Purtroppo, questo è stato a volte frainteso come una promessa firmata col sangue, piuttosto che un serio impegno del Team a "darci dentro". Per evitare questa confusione, l'obiettivo dello sprint è ora chiamato una "previsione", che viene comunicata al Product Owner.

Scrum incoraggia chi lavora ad acquisire diverse competenze, piuttosto che solo "lavorare in base al proprio titolo", come ad esempio un "tester" che fa solo test. In altre parole, i membri del Team "vanno dove c'è lavoro da fare" e aiutano il più possibile. Se ci sono molte attività di test, allora *tutti* i membri del Team possono aiutare. Questo non implica che tutti siano generalisti, senza dubbio alcune persone sono particolarmente abili in fase di test (e così via), ma i membri del Team lavorano insieme e acquisiscono nuove competenze gli uni dagli altri.

Di conseguenza, durante la generazione e la stima dei compiti nello Sprint Planning, non è necessario - né opportuno - che i membri del Team si offrano volontari per tutti i compiti "che possono fare meglio". Piuttosto, è meglio offrirsi per un solo compito alla volta, quando è il momento di prendere una nuova attività, e considerare di scegliere attività che implicheranno apprendimento (ad esempio lavorando in parallelo con uno specialista). Questo è uno dei motivi per non pre-assegnare compiti durante lo Sprint Planning, anzi questo dovrebbe essere fatto solo in base alla necessità durante lo Sprint.

Ciò detto, ci sono *rari* casi in cui *John* può svolgere un compito particolare, perché ci vorrebbe troppo tempo o sarebbe impossibile per gli altri imparare - forse John è l'unica persona con qualche abilità artistica nel disegnare immagini. Gli altri membri del Team non sarebbero in grado di disegnare un

"omino stilizzato" neanche se da ciò dipendesse la loro vita. In questo raro caso - e se non è raro e non diventa sempre più raro man mano che il Team impara, c'è qualcosa di sbagliato - può essere necessario chiedersi se il totale delle attività di disegno pianificate che *devono* essere svolte da John sono realizzabili nel breve tempo di uno Sprint.

Molti Team hanno uno Sprint Backlog nella forma di un foglio delle attività applicato a parete (spesso chiamato uno **Scrum Board**) dove i compiti (scritti su post-it) migrano durante lo Sprint tra le colonne etichettate "Da fare", "Lavoro in corso" e "Fatto". Vedi Figura 5.

Figura 5. Visual management – Compiti dello Sprint Backlog a parete



Uno dei pilastri di Scrum è che una volta che il Team fissa l'obiettivo per lo Sprint, eventuali integrazioni o modifiche devono essere differiti fino al prossimo Sprint. Ciò significa che se a metà dello Sprint il Product Owner decide che c'è una nuova voce che vorrebbe assegnare al Team, non si può fare il cambiamento fino all'inizio dello Sprint successivo. Se una circostanza esterna sembra cambiare in modo significativo le priorità, al punto che il Team starebbe sprecando il suo tempo se continuasse a lavorarci, il Product Owner e il Team possono terminare lo Sprint. Il Team si ferma, e un nuovo Sprint Planning Meeting fa iniziare un nuovo Sprint. La perturbazione causata da questa operazione è di solito significativa; questo serve come disincentivo verso il Product Owner o il Team a ricorrere a questa decisione drastica.

Vi è una forte influenza positiva che arriva al Team quando è protetto da cambiamenti dell'obiettivo durante lo Sprint. In primo luogo, il Team si mette al lavoro sapendo con assoluta certezza che il suo obiettivo non cambierà, il che rafforza l'attenzione del Team a garantirne il completamento. In secondo luogo, esso disciplina il Product Owner a pensare veramente alla priorità assegnata alle voci sul Product Backlog e offerte al Team per lo Sprint.

Seguendo queste regole di Scrum il Product Owner si avvantaggia sotto due aspetti. In primo luogo, ha la fiducia di sapere che il Team si è impegnato a fare del suo meglio per completare un insieme realistico e chiaro di lavori che esso stesso ha scelto. Nel corso del tempo un Team può diventare molto abile nello scegliere e fornire una previsione realistica. In secondo luogo, il Product Owner è spinto a fare tutte le modifiche che vuole al Product Backlog prima dell'inizio del *prossimo* Sprint.

A quel punto, aggiunte, cancellazioni, modifiche, e ri-assegnazioni di priorità sono tutte possibili e accettabili. Sebbene il Product Owner non sia in grado di apportare modifiche agli elementi selezionati e in fase di sviluppo durante l'attuale Sprint, ha davanti a sé solo il tempo della durata di uno Sprint, o anche meno, per poter apportare le modifiche che desidera. Lo stigma intorno al cambiamento è superato - cambiamento di direzione, cambiamento dei requisiti, o semplicemente cambiamento di opinione - e può essere per questo motivo che i Product Owners sono di solito più entusiasti di Scrum di chiunque altro.

Daily Scrum

Riassunto: Aggiornamento e coordinamento tra i membri del Team.

Partecipanti: Il Team; Product Owner (se richiesto); ScrumMaster (di solito è presente e comunque assicura che il Team tenga il meeting)

Durata: 15 minuti al massimo.

Una volta che lo Sprint è iniziato, il Team si impegna in un'altra delle pratiche chiave di Scrum: Il **Daily Scrum**. Questo è un breve incontro (15 minuti o meno) che avviene ogni giorno lavorativo all'ora stabilita. Tutti i membri del Team partecipano. Per mantenerlo di breve durata, si raccomanda che tutti rimangano in piedi. E' un'opportunità per il Team di sincronizzare il proprio lavoro e informarsi reciprocamente sugli ostacoli. Nel Daily Scrum, uno per uno, ogni membro del gruppo riferisce su tre cose *agli altri membri del Team*: (1) Che cosa è stato fatto dopo l'ultima riunione, (2) Che cosa sarà fatto prima della prossima riunione e (3) Quali ostacoli ci sono da affrontare.

Si noti che il Daily Scrum non è una riunione di stato per riportare ad un manager, è un po' di tempo in cui un Team auto-gestito condivide al suo interno quello che sta succedendo, per contribuire a coordinarsi. Qualcuno rende noti eventuali blocchi, e lo ScrumMaster è responsabile per aiutare i membri del Team a risolverli. Non c'è discussione approfondita durante il Daily Scrum, il tema è di *riportare* le risposte alle tre domande, se è necessario la discussione avviene subito dopo il Daily Scrum in una o più riunioni di follow-up in parallelo, anche se in Scrum non si è tenuti a parteciparvi. Un incontro di follow-up è un evento comune in cui alcuni o tutti i membri del Team si adeguano alle informazioni che hanno sentito nel Daily Scrum: in altre parole, un altro ciclo di ispezione e adeguamento.

Per i Team nuovi in Scrum, si raccomanda in genere che il Daily Scrum *non* venga frequentato da manager o altre posizioni percepite come di autorità. Questo pone il rischio che il Team si senta "monitorato" e che senta la pressione di segnalare importanti progressi ogni giorno (una aspettativa non realistica), e si senta inibito a segnalare problemi - inoltre tende a minare l'autogestione del Team, e favorire la microgestione. Sarebbe più utile invece, per uno stakeholder, raggiungere il Team fuori dalla riunione, e proporsi di aiutare a sbloccare le situazioni che frenano il progresso del Team..

Tracciare l'avanzamento durante lo Sprint

Il Team in Scrum si gestisce da solo, e per farlo con successo, è necessario che sappia come sta procedendo. Ogni giorno, i membri del Team aggiornano la loro stima dello sforzo rimanente per completare il loro lavoro attuale nello **Sprint Backlog** (Figura 6). E' pratica comune che qualcuno tracci lo sforzo restante per il Team nel suo complesso, e disegnarlo sul grafico dello **Sprint Burndown** (Figura 7 e Figura 8).

Questo grafico mostra, ogni giorno, una nuova stima di quanto lavoro rimane fino a quando il Team avrà terminato. Idealmente, questo è un grafico che è inclinato *verso il basso* su una traiettoria per raggiungere "zero sforzo residuo" entro l'ultimo giorno dello Sprint. Quindi viene chiamato un grafico *burndown*. E mentre a volte sembra buono, spesso non è così, questa è la realtà dello sviluppo di un prodotto.

La cosa importante è che mostra al Team l'avanzamento verso il loro obiettivo, non in termini di quantità di tempo che è stato già *speso* (un fatto irrilevante per l'*avanzamento*), ma in termini di quantità di lavoro che *rimane da fare* - cosa separa il Team dal proprio obiettivo. Se il grafico burndown non sta tendendo in basso verso il completamento alla fine dello Sprint, allora il Team ha bisogno di regolarsi, in modo da ridurre i contenuti del lavoro o di trovare un modo per lavorare in modo più efficace, pur mantenendo un ritmo sostenibile.

Mentre il grafico Sprint Burndown può essere creato e visualizzato utilizzando un foglio di calcolo, molti Team trovano più efficace mostrarlo su un cartellone su una parete del loro ufficio, con

aggiornamenti scritti a mano: questa soluzione "low-tech / high-touch" è veloce, semplice, e spesso più visibile di un grafico a computer.

		Nuove stime dello sforzo Rimanente alla fine del giorno...							
Voce del Product Backlog	Sprint Task	Volontario	Stima iniziale dello sforzo	1	2	3	4	5	6
Come acquirente, voglio inserire un libro nel carrello d'acquisto	Modificare il database	Gianni	5	4	3	0	0	0	
	Creare la pagina web (UI)	Jing	3	3	3	2	0	0	
	Creare la pagina web (Javascript logic)	Tracy & Sam	2	2	2	2	1	0	
	Scrivere i test di accettazione automatici	Sara	5	5	5	5	5	0	
	Aggiornare la pagina di aiuto per l'acquirente	Gianni & Jing	3	3	3	3	3	0	
	...								
Migliorare le prestazioni della elaborazione delle transazioni	Inserire il codice DCP e completare i test al livello-layer		5	5	5	5	5	5	
	completare la procedura ordini per pRank		3	3	8	8	8	8	
	adattare il DCP e il reader a pRank http API		5	5	5	5	5	5	
...									
Total			50	49	48	44	43	34	

Figura 6. Aggiornamenti quotidiani di lavoro rimanente sullo Sprint Backlog

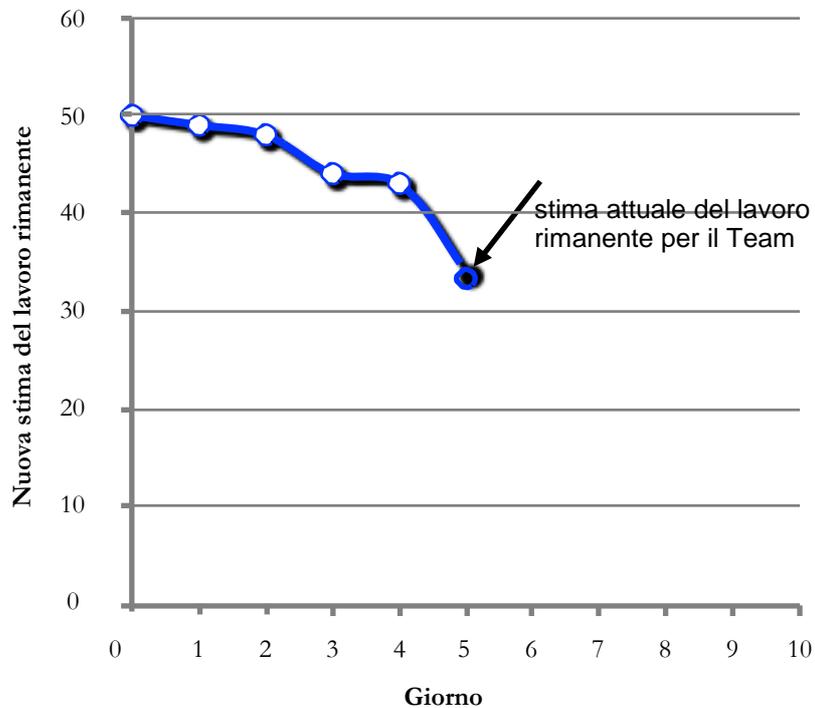


Figura 7. Grafico di Sprint Burndown

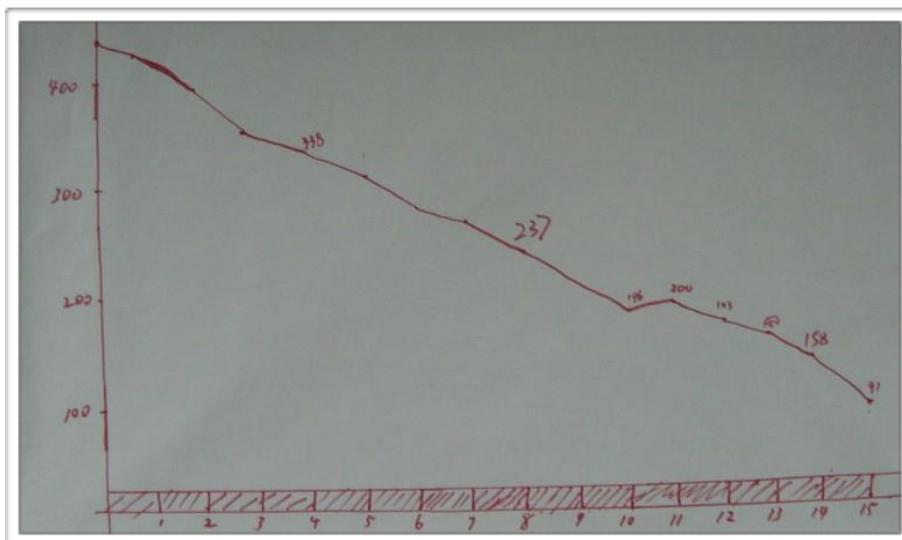


Figura 8. Visual Management: Grafico Sprint Burndown disegnato a mano

Raffinamento del Product Backlog

Riassunto: Suddivisione delle voci più grandi, analisi delle voci, nuova stima, e nuova assegnazione della priorità, per Sprint futuri.

Partecipanti: Il Team e il Product Owner saranno presenti per l'intera attività se sono gli esperti che possono aiutare ad affinare i dettagli, altrimenti possono partecipare solo ad una parte per impostare la direzione o ridefinire le priorità; possono partecipare altri soggetti che capiscano i requisiti e possono aiutare il Team; lo ScrumMaster sarà presente durante le sessioni iniziali per abituare il gruppo ad essere efficace, altrimenti possono non partecipare.

Durata: Di solito, non più del 10% della capacità del Team per lo Sprint, anche se può essere necessario più tempo per voci che richiedono "analisi pesante". Ad esempio, in Sprint di due settimane, forse un giorno viene speso per l'affinamento.

Una delle linee guida di Scrum meno conosciute, ma preziose, è che una certa percentuale di ogni Sprint dovrebbe essere dedicata da tutto il Team all'affinamento (o "grooming") del Product Backlog per supportare Sprint futuri. Ciò comprende l'analisi dettagliata dei requisiti, dividendo voci di grandi dimensioni in voci più piccole, la stima di nuove voci, e la nuova stima di quelle esistenti. Scrum non si esprime su come questo lavoro vada fatto, ma una tecnica utilizzata frequentemente è un workshop dedicato da svolgersi vicino alla metà o alla fine dello Sprint, in modo che il Team, il Product Owner e gli altri soggetti interessati si possano dedicare a questo lavoro senza interruzioni.

Questa attività *non* è per affinare voci selezionate per lo Sprint in corso; è fatta per voci da realizzare in futuro, molto probabilmente nei prossimi due Sprint. Con questa pratica, lo Sprint Planning diventa relativamente semplice, perché il Product Owner e il Team di Scrum iniziano la pianificazione a partire da un insieme di voci, ben analizzate e accuratamente stimate. Il segno che questo workshop di affinamento non è stato fatto (o non viene fatto bene) è che nello Sprint Planning si presentano questioni significative, ci sono nuove scoperte, confusione e sensazione di incompletezza; il lavoro di pianificazione quindi spesso sconfinava nel corso dello Sprint stesso, cosa che non è in genere auspicabile.

Sprint Review

Riassunto: Ispezione e adeguamento relativi all'incremento di funzionalità del prodotto.

Partecipanti: Team, Product Owner, ScrumMaster, altri soggetti interessati se opportuno, se invitati dal Product Owner.

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

Dopo il termine dello Sprint, c'è la **Sprint Review**, dove si passa in rassegna lo Sprint. Presenti a questo

incontro sono il Product Owner, i membri del Team, lo ScrumMaster, ed inoltre clienti, utenti, soggetti interessati, esperti, i dirigenti, e chiunque altro sia interessato. Per uno Sprint di due settimane la lunghezza massima è di due ore. Chiunque sia presente è libero di porre domande e dare input.

La Review è spesso erroneamente chiamata la "demo", ma questo nome non coglie il vero intento di questo incontro. L'idea chiave in Scrum è *ispezionare e adeguare*, per vedere e imparare che cosa sta succedendo e poi evolvere in base al feedback, in un ciclo che si ripete. La Sprint Review è un'attività di ispezione e adeguamento del *prodotto*. E' un tempo dedicato dal Product Owner ad imparare come stanno evolvendo il prodotto ed il Team (cioè una revisione dello Sprint) e dedicato dal Team ad imparare come stanno evolvendo il Product Owner e il mercato.

Di conseguenza, l'elemento critico della Review è una *conversazione* approfondita tra il Team e il Product Owner per apprendere la situazione, per ottenere consigli, e così via. La Review comprende sicuramente l'utilizzo del software vero e proprio, che il Team ha costruito durante lo Sprint, ma se il focus della revisione fosse solo di guardare al prodotto piuttosto che parlarne, si creerebbe uno squilibrio.

La parte "software dal vivo" della Sprint Review non è una "presentazione" che dà il Team – non ci sono slides. Essa è destinata ad essere una ispezione pratica del software reale in esecuzione, per esempio, in un ambiente di sviluppo sandbox. Ci saranno uno o più computer nella sala della Review su cui le persone possono controllare e utilizzare il software in tempo reale. Preferite una sessione attiva in cui gli utenti reali e il Product Owner interagiscono con il software, piuttosto che una demo passiva da parte del Team.

L'obiettivo è spendere non più di 30 minuti per la preparazione della Sprint Review, altrimenti significa che qualcosa non va.

Retrospettiva di Sprint

Riassunto: Ispezione e adeguamento riguardanti il processo e l'ambiente.

Partecipanti: Team, ScrumMaster, il Product Owner se richiesto. Altri soggetti interessati possono essere invitati dal Team, non sono altrimenti autorizzati a partecipare.

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

Con la Sprint Review si ispeziona e si adegua ciò che riguarda il *prodotto*. Con la **Retrospettiva di Sprint**, che segue la Sprint Review, si ispeziona e si adegua ciò che riguarda il *processo e l'ambiente*. E' l'occasione per il Team per discutere di ciò che funziona e ciò che non funziona, e concordare le modifiche da provare. A volte lo ScrumMaster può agire da moderatore efficace per la retrospettiva, ma può essere meglio trovare un soggetto neutrale per facilitare l'incontro, un buon approccio è che ogni ScrumMaster vada a moderare retrospettive degli altri Team, ciò permette l'impollinazione incrociata tra i Team.

Ci sono molte tecniche per condurre una Retrospettiva, e il libro *Agile Retrospectives* (Derby, Larsen 2006) offre un catalogo utile di tecniche.

Molti Team tengono retrospettive solo concentrandosi su *problemi*, e questo è decisamente da evitare. Infatti può portare a far vedere le retrospettive, alle persone coinvolte, come eventi un po' deprimenti o negativi. Invece, bisogna assicurarsi che ogni retrospettiva si concentri anche su aspetti positivi e punti di forza, ci sono diversi libri sulla *inchiesta elogiativa* che offrono consigli più dettagliati.

Retrospettive che usano sempre la stessa tecnica di analisi possono diventare noiose, quindi, meglio introdurre varie tecniche con l'andare del tempo.

Iniziare il prossimo Sprint

Dopo la Sprint Review, il Product Owner può aggiornare il Product Backlog con qualsiasi nuova revisione: l'aggiunta di nuovi elementi, eliminando quelli obsoleti, o modificare quelli esistenti. Il Product Owner è responsabile di assicurare che questi cambiamenti si riflettano nel Product Backlog. Vedi Figura 9 per un esempio del Product Backlog aggiornato.

		Nuove stime allo Sprint ...							
Priorità	Voce	Dettagli (wiki URL)	Stima iniziale dimensioni	1	2	3	4	5	6
1	Come acquirente, voglio inserire un libro nel carrello d'acquisto (vedi esempi UI sulla wiki page)	...	5	0	0	0			
2	Come acquirente, Voglio rimuovere un libro dal carrello d'acquisto	...	2	0	0	0			
3	Migliorare le prestazioni del processo di transazione (vedi le metriche sui target di prestazioni sul wiki)	...	13	13	0	0			
4	Investigare soluzioni per velocizzare la validazione della carta di credito	...	20	20	20	0			
5	Aggiornare tutti i server ad Apache 2.2.3	...	13	13	13	13			
6	Eseguire diagnosi e correggere gli errori dello script di elaborazione dell'ordine (bugzilla ID 14823)	...	3	3	3	3			
7	Come acquirente voglio creare e salvare una lista dei desideri	...	40	40	40	40			
8	Come acquirente, voglio aggiungere o cancellare voci nella mia lista dei desideri	...	20	20	20	20			
...		
			537	580	570	500			

Figura 9. Product Backlog aggiornato

Non ci sono pause tra Sprint – I Team normalmente passano da una Retrospettiva il pomeriggio ad un prossimo Sprint Planning la mattina seguente (o dopo il fine settimana).

Uno dei principi dello sviluppo agile è "ritmo sostenibile", e solo lavorando regolarmente ma per un numero di ore di lavoro ragionevole può consentire ai Team di continuare questo ciclo all'infinito. La produttività cresce nel tempo attraverso l'evoluzione delle pratiche del Team, e la rimozione degli ostacoli alla produttività del Team, non attraverso superlavoro o compromessi sulla qualità.

Gli Sprint continuano fino a quando il Product Owner decide che il prodotto è pronto per il rilascio finale. La visione perfetta di Scrum è che il prodotto sia potenzialmente rilasciabile alla fine di ogni sprint, il che implica non sia necessario altro lavoro di riepilogo, come ulteriore testing o documentazione. L'implicazione è che *tutto* sia completamente *finito* ad ogni Sprint, e che si potrebbe effettivamente rilasciare e consegnare immediatamente dopo la Sprint Review.

Tuttavia, molte organizzazioni hanno pratiche di sviluppo, strumenti e infrastrutture deboli, non possono raggiungere la realizzazione di questa visione perfetta e quindi ci sarà la necessità di una "Release Sprint" per gestire del lavoro rimanente. Quando è necessaria una "Release Sprint", si può ritenere un male necessario, e il lavoro dell'organizzazione è quello di migliorare le proprie pratiche, in modo che questo non sia più necessario.

Gestire le Release

Una domanda che a volte sorge è come, in un modello iterativo, si può fare pianificazione delle release a lungo termine. Ci sono due casi da considerare: (1) un nuovo prodotto nella sua prima versione, e (2) un prodotto esistente in una versione successiva.

Nel caso di un nuovo prodotto o di un *prodotto esistente che adotta Scrum per la prima volta*, vi è la necessità di fare un primo affinamento del Product Backlog prima del primo Sprint, dove il Product Owner e il Team formano un vero e proprio Scrum Product Backlog. Questo potrebbe richiedere alcuni giorni o una settimana, e comporta un workshop (a volte chiamato Creazione Iniziale del Product Backlog o Release Planning), dettagliata analisi di alcuni requisiti, e la stima di tutte le voci individuate per la prima release.

Sorprendentemente in Scrum, nel caso di un prodotto stabile con un Product Backlog consolidato, non dovrebbe esserci la necessità di alcuna pianificazione speciale o estesa per il prossimo rilascio. Perché? Il motivo è che il Product Owner e il Team dovrebbero affinare il Product Backlog ad ogni Sprint (cinque o dieci per cento di ogni Sprint), perpetuando la preparazione per il futuro. Questo *modo continuo di sviluppare il prodotto* evita la necessità di una problematica sequenza di preparazione-esecuzione-conclusione a tappe successive come si vede nello sviluppo tradizionale con ciclo di vita sequenziale.

Durante un workshop di affinamento iniziale del Product Backlog e durante l'affinamento continuo ad ogni Sprint, il Team e il Product Owner faranno pianificazione della release, affinando le stime, le priorità e il contenuto, mentre acquisiscono maggiori conoscenze.

Alcune release sono vincolate ad una data; ad esempio: "Noi rilasceremo la versione 2.0 del nostro progetto a quel trade-show il 10 novembre." In questa situazione, il Team completerà il maggior numero di sprint (e inserirà il maggior numero possibile di funzionalità), nel tempo a disposizione. Altri prodotti richiedono che determinate funzionalità siano realizzate prima di poter essere considerati completi, e il prodotto non può essere lanciato fino a quando questi requisiti non sono soddisfatti, per quanto tempo ciò possa prendere. Poiché Scrum punta sul produrre codice potenzialmente rilasciabile ad ogni Sprint, il Product Owner può scegliere di iniziare con delle interim release, per consentire al cliente di iniziare a cogliere prima i vantaggi del lavoro completato.

Dal momento che non si può sapere tutto in anticipo, l'obiettivo è di creare e perfezionare un piano per dare alla release il più ampio senso, e chiarire come verranno prese le decisioni rischiose (ambito versus calendario, per esempio). Pensatela come la tabella di marcia che vi guida verso la vostra destinazione finale; quali strade esattamente scegliere e quali decisioni prendere durante il viaggio possono essere determinate strada facendo.

La destinazione è più importante del viaggio.

La maggior parte dei Product Owner scelgono una strategia di rilascio. Ad esempio, saranno loro a decidere una data di uscita, e lavoreranno con il Team per stimare le voci di Product Backlog che possono essere completate entro tale data. Le voci che sono previste essere nella versione in corso sono a volte chiamati gli *articoli di rilascio*. In situazioni in cui è previsto un impegno "prezzo fisso / data fissata / contenuti fissi" - per esempio, sviluppo a contratto - uno o più di questi parametri deve prevedere un buffer per consentire incertezze e cambiamenti; a questo riguardo, Scrum non è diverso da altri approcci.

Focalizzarsi sull'Applicazione o sul Prodotto

Per le applicazioni o prodotti - sia da mettere sul mercato sia per uso interno ad un'organizzazione - Scrum sposta i gruppi di lavoro dal vecchio modello *progetto*-centrico verso un modello di *sviluppo di applicazione/prodotto continuo*. Non c'è più un progetto con un inizio, un "nel mezzo" e una fine. E quindi, non c'è un project manager tradizionale.

Piuttosto, c'è semplicemente un Product Owner stabile e un Team di lunga durata e auto-gestito che collaborano in una serie "senza fine" di Sprint a lunghezza fissa, fino a quando il prodotto o l'applicazione è a fine vita. Tutta la necessaria gestione del "progetto" è nelle mani del Team e del Product Owner - che è un cliente interno all'azienda o viene dal Product Management. Non è gestito da un responsabile IT o da qualcuno che è parte di un Project Management Office.

Scrum può essere utilizzato anche per veri *progetti* che sono iniziative con unico rilascio (piuttosto che attività per creare o sviluppare applicazioni di lunga durata); ancora, in questo caso il Team e il Product Owner fanno la gestione del progetto.

Cosa fare se una o più applicazioni esistenti non richiedono nuovo lavoro sufficiente per giustificare un Team dedicato per lungo tempo per ogni applicazione? In questo caso, un Team stabile può assegnarsi voci da un'applicazione in uno Sprint, e poi voci da un'altra applicazione nello Sprint successivo, in questa situazione gli Sprint sono spesso molto brevi, ad esempio una settimana.

Di tanto in tanto, c'è insufficiente nuovo lavoro anche dall'applicazione principale, e il Team può assegnarsi voci da *diverse* applicazioni nello stesso Sprint; tuttavia fate attenzione a questa soluzione in quanto può evolvere in multitasking improduttivo tra più applicazioni. Un tema di produttività di base in Scrum è che il Team si possa *focalizzare* su un prodotto o su un'applicazione durante uno Sprint.

Sfide Comuni

Scrum non è solo un insieme concreto di prassi - piuttosto, e cosa più importante, è un ambiente di lavoro che fornisce trasparenza e un meccanismo che permette di "ispezionare ed adeguare". Scrum funziona rendendo visibili le disfunzioni e gli impedimenti che stanno influenzando l'efficacia del Product Owner e del Team, in modo che possano essere affrontati e risolti. Ad esempio, il Product Owner può non conoscere realmente il mercato, le funzionalità, o come stimare il loro valore relativo di business. Oppure il Team può non essere capace di stimare lo sforzo o non essere preparato nel lavoro di sviluppo.

L'impianto di Scrum rivelerà rapidamente queste debolezze. Scrum non risolve i problemi dello

sviluppo, ma li rende dolorosamente visibili, e fornisce un quadro di riferimento per le persone nell'esplorare i modi per risolvere i problemi in cicli brevi e con piccoli esperimenti di miglioramento.

Supponiamo che il Team non riesca a consegnare ciò che è previsto nel primo Sprint a causa di una debole attività di analisi e scarsa abilità nella stima. Per il Team, questo è visto come un fallimento. Ma in realtà, questa esperienza è il primo passo necessario per far diventare le proprie previsioni più realistiche e meditate. Questo modello - di Scrum che contribuisce a rendere la disfunzione visibile, permettendo al Team di fare qualcosa al riguardo - è il meccanismo di base che produce i vantaggi più significativi sperimentati dai Team che usano Scrum.

Un errore che si fa comunemente, quando adottare Scrum si presenta come impresa impegnativa, è quello di cambiare Scrum. Ad esempio, Team che hanno difficoltà a consegnare potrebbero decidere di rendere la durata dello Sprint estensibile, in modo che non si esaurisce mai il tempo a disposizione - e nel processo, non assicurarsi che si debba imparare a fare un lavoro migliore nello stimare e gestire il proprio tempo. In questo modo, senza il coaching e il supporto di uno ScrumMaster esperto, le organizzazioni possono modificare Scrum in ciò che non è altro che l'immagine speculare delle proprie debolezze e disfunzioni, e minare il vero vantaggio che Scrum offre: rendere visibile ciò che è bene e ciò che è male, e dando all'organizzazione la scelta di elevarsi ad un livello superiore.

Un altro errore comune è quello di ritenere che una pratica sia sconsigliata o vietata solo perché Scrum non la prevede esplicitamente. Ad esempio, Scrum non richiede al Product Owner di impostare una strategia a lungo termine per il suo prodotto, né richiede ai tecnici di chiedere il parere di ingegneri più esperti sui problemi tecnici complessi. Scrum lascia agli individui coinvolti di prendere la decisione giusta, e nella maggior parte dei casi, entrambe queste pratiche (insieme a molte altre) sono certamente consigliate.

Un'altra cosa di cui diffidare sono i manager che impongono Scrum ai loro Team; Scrum consiste nel dare spazio e strumenti ad un Team per gestire se stesso, e quando ciò è dettato dall'alto non è una ricetta per il successo. Un approccio migliore potrebbe iniziare con un Team Scrum che impara da un altro Team o da un manager, ricevendo una formazione professionale completa; il Team poi prende la decisione se seguire o meno le pratiche fedelmente per un periodo definito; al termine di tale periodo, il Team valuterà la sua esperienza, e deciderà se continuare.

La buona notizia è che, mentre il primo sprint è di solito molto impegnativo per il Team, i benefici di Scrum tendono ad essere visibili prima della sua fine, portando molti nuovi Scrum Team ad esclamare: "Scrum è impegnativo, ma sicuramente è molto meglio di quanto stavamo facendo prima!"

Appendice A: Letture aggiuntive

C'è una gran quantità di materiale pubblicato su Scrum. In questa sezione sul materiale di riferimento, vorremmo suggerire del materiale supplementare on-line e alcuni libri.

Materiale Online:

- [The Lean Primer - An introduction to Lean Thinking, an important influence to Scrum.](http://www.leanprimer.com)
<http://www.leanprimer.com>
- [The Distributed Scrum Primer - Additional tips for teams who aren't co-located.](http://www.goodagile.com/distributedscrumprimer/)
<http://www.goodagile.com/distributedscrumprimer/>
- [The ScrumMaster Checklist - A list of question that good ScrumMasters use.](http://www.scrummasterchecklist.org/)
<http://www.scrummasterchecklist.org/>
- [Feature Team Primer - Scaling Scrum with Feature Teams,](http://www.featureteams.org)
<http://www.featureteams.org>
- [The Agile Atlas - Core Scrum. ScrumAlliance description of Scrum.](http://agileatlas.org/atlas/scrum)
<http://agileatlas.org/atlas/scrum>
- [Scrum Guide - Scrum.org description of Scrum.](http://www.scrum.org/Scrum-Guides)
<http://www.scrum.org/Scrum-Guides>
- [Agile Contracts Primer - How to make Scrum-friendly contracts.](http://www.agilecontracts.org/)
<http://www.agilecontracts.org/>

Libri:

- Leading Teams - Richard Hackman
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum - Craig Larman, Bas Vodde
- Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum - Craig Larman, Bas Vodde
- Agile Project Management with Scrum - Ken Schwaber
- Succeeding with Agile: Software Development using Scrum - Mike Cohn

Appendice B: Terminologia

Burn Down

L'andamento del lavoro che rimane da svolgere lungo l'arco di tempo di uno Sprint, una release o un prodotto. La fonte dei dati grezzi è lo Sprint Backlog o il Product Backlog. Il lavoro che rimane da fare viene tracciato sull'asse verticale ed i periodi di tempo (giorni di uno Sprint, o i diversi Sprint) sono tracciati sull'asse orizzontale.

Daily Scrum (Scrum Quotidiano)

Una breve riunione che giornalmente è tenuta da ciascun Team durante la quale i membri del Team ispezionano il loro lavoro, sincronizzano il loro lavoro e il relativo avanzamento e riportano gli ostacoli allo ScrumMaster per la loro rimozione.

Possono seguire al Daily Scrum altre riunioni per adattare il lavoro in arrivo, per ottimizzare lo Sprint.

Done (Fatto)

Completare come reciprocamente concordato da tutte le parti in modo conforme agli standard, alle convenzioni ed alle linee guida di un'organizzazione. Quando qualcosa viene segnalato come "fatto" in occasione della riunione di Sprint Review, deve esserlo in modo conforme alla definizione di "fatto" concordata.

Incremento

Funzionalità del prodotto che viene sviluppata dal Team durante uno Sprint con la caratteristica di essere potenzialmente rilasciabile o utilizzabile dagli stakeholders del Product Owner.

Incremento di Funzionalità Potenzialmente Rilasciabile

Una parte completa dell'intero prodotto o di un sistema che potrebbe essere utilizzata dal Product Owner o dagli stakeholders che hanno scelto per la sua implementazione.

Lavoro Rimanente Stimato (voci dello Sprint Backlog)

Il numero di ore che un membro del Team stima debbano ancora essere lavorate su una qualsiasi attività. Questa stima viene aggiornata al termine di ogni giorno, quando l'attività viene elaborata sullo Sprint Backlog. La stima è il totale sforzo residuo stimato, indipendentemente dal numero di persone che eseguono il lavoro.

Product Backlog

Una lista di requisiti con priorità assegnata e tempi stimati per trasformarli in funzionalità del prodotto completate. Le stime sono più precise tanto più alta è la priorità di un elemento nel Product Backlog. L'elenco nasce e cambia, cambiando le condizioni di mercato o la tecnologia.

Product Owner

La persona responsabile della gestione del Product Backlog in modo da massimizzare il valore del prodotto. Il Product Owner è responsabile di rappresentare gli interessi di tutti quelli che hanno un tornaconto dal progetto e dal suo prodotto risultante.

Riunione di Retrospettiva dello Sprint

Un incontro moderato dallo ScrumMaster in cui il Team al completo discute lo Sprint appena concluso e determina quello che potrebbe essere cambiato, che potrebbe rendere il prossimo Sprint più divertente e produttivo.

Riunione di Sprint Planning

Un incontro con limite di durata a quattro ore (in relazione ad uno Sprint di due settimane) che avvia ogni Sprint. L'incontro è suddiviso in due segmenti di due ore, anch'essi con limite di durata. Durante la prima parte il Product Owner presenta le voci di massima priorità del Product Backlog al Team. Il Team e il Product Owner collaborano a far sì che il Team possa determinare quanta parte del Product Backlog può trasformarsi in funzionalità durante il prossimo Sprint. Durante la seconda parte, il Team pianifica come raggiungere questo obiettivo attraverso la progettazione e scomposizione del lavoro in

modo da capire come raggiungere il traguardo dello Sprint.

Riunione di Sprint Review

Un incontro con limite di durata a due ore (in relazione ad uno Sprint di due settimane) alla fine di ogni Sprint dove il Team collabora con il Product Owner e gli stakeholders nel verificare i risultati dello Sprint.

Di solito si inizia con una rassegna degli elementi completati del Product Backlog, una discussione su opportunità, vincoli e rischi, e su quali potrebbero essere le cose migliori da fare nello Sprint successivo (potenzialmente causando cambiamenti al Product Backlog). Solo le funzionalità del prodotto completate possono essere dimostrate.

Scrum

Non è un acronimo, ma il meccanismo del gioco del rugby per rimettere in gioco la palla.

ScrumMaster

La persona responsabile del processo Scrum, della sua corretta attuazione e della massimizzazione dei suoi benefici.

Sprint

Un'iterazione, o una ripetizione di cicli di lavoro analoghi, che produce incremento del prodotto o di un sistema.

Non deve durare più di un mese e di solito dura più di una settimana. La durata rimane fissata durante tutto il lavoro complessivo e per tutti i Team che lavorano sullo stesso sistema o prodotto la lunghezza dello Sprint è la medesima.

Sprint Backlog

Un elenco di attività che il Team svolge nell'ambito di uno Sprint. Questo è spesso scomposto in una serie di compiti più dettagliati. L'elenco emerge durante lo Sprint Planning e può essere aggiornato dal Team durante lo Sprint con compiti rimossi o nuovi compiti aggiunti, se necessario. Ogni attività nello Sprint Backlog verrà tracciata durante lo Sprint e mostrerà lo sforzo residuo stimato.

Sprint Backlog Task

Uno dei compiti che il Team o un suo membro stabilisce sia richiesto per trasformare gli elementi concordati del Product Backlog nella funzionalità del sistema.

Stakeholder (soggetto interessato)

Qualcuno interessato all'esito di un progetto, sia perché l'ha finanziato, perché lo userà, o perché ne subirà gli effetti.

Team

Un gruppo multifunzionale di persone che ha il compito di gestire se stesso per sviluppare un incremento di prodotto ad ogni Sprint.

Team di sviluppo

Un altro nome per il ruolo del Team.

Time box

Un periodo di tempo che non può essere superato e all'interno del quale si verifica un evento o una riunione. Ad esempio, un incontro di Daily Scrum è time-boxed a quindici minuti e termina alla fine dei quindici minuti, in ogni caso. Per quanto riguarda le riunioni, queste potrebbero durare di meno. Anche la lunghezza di uno Sprint è limitata a quanto fissato.

Voce del Product Backlog

I requisiti funzionali, i requisiti non funzionali, le questioni prioritarie in ordine di importanza secondo il business e secondo le interdipendenze, con associata una stima. La precisione della stima dipende dalla priorità e dalla granularità della voce nel Product Backlog, con gli articoli a più alta priorità che potranno essere selezionati per il prossimo Sprint quando sono molto affinati e precisi.