

Pacchetto di Espansione della Guida a Scrum

Ralph Jocham

John Coleman

Jeff Sutherland

2026-01-18T09:00:00Z

Risorse raccolte per il Pacchetto di Espansione della Guida a Scrum

Questo documento è una raccolta di opere indipendenti. Ogni sezione mantiene la sua licenza originale o lo stato del copyright, come indicato. Si prega di fare riferimento a ciascuna sezione per specifici diritti e requisiti di utilizzo.

Fonte: Guida a Scrum 2020, Pacchetto di Espansione della Guida a Scrum

Licenza: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).

© 2025 Ralph Jocham, John Coleman, and Jeff Sutherland.

Avviso di modifica: Questo è un adattamento dell'originale Guida a Scrum 2020 e si notifica che sono state apportate modifiche all'originale.

Esclusione di responsabilità: Nessuna garanzia è fornita, le nozioni contenute nel presente documento possono essere usate a proprio rischio e pericolo.

Questa sezione è offerta ai sensi della licenza Attribution-ShareAlike 4.0 International di Creative Commons.

Usando questo Pacchetto di Espansione della Guida a Scrum, accetti i termini della licenza CC BY-SA 4.0.

Contesto di riferimento

L'adozione di Scrum da parte di Jeff Sutherland nel 1993 presso Easel è stata ispirata dagli articoli di Christopher Langton [1-2] sulla teoria dei Sistemi Complessi Adattivi (CAS) [3-6] dei Los Alamos Labs, che dimostravano che i sistemi evolvono più velocemente ai margini del caos.

Ken Schwaber e Jeff Sutherland hanno guidato lo sviluppo del framework Scrum. La Guida a Scrum 2020 [7], incluso il contributo di Ralph Jocham [8], descrive gli elementi essenziali di Scrum. A Simple Guide to Scrum di Tobias Mayer [9] è una versione abbreviata e la Scrum Hexis [10] elabora la Guida a Scrum 2020 da una prospettiva del 2025. Questo documento spiega molto su Scrum e offre indicazioni per i tempi attuali, ma alla fine diverge in alcuni punti. Le principali divergenze dalla Guida Scrum del 2020 sono indicate in questo modo (a volte con una spiegazione).

Scopo del Pacchetto di Espansione della Guida a Scrum

Questo Pacchetto di Espansione della Guida a Scrum (SGEP) spiega il *cosa* e il *perché* di ogni elemento di Scrum alla luce delle esigenze attuali. Ogni elemento ha uno scopo specifico e contribuisce al valore complessivo e ai risultati ottenuti con Scrum. Questo Pacchetto di Espansione verrà aggiornato regolarmente. È fortemente consigliato che il lettore legga il documento in modo sequenziale, almeno la prima volta.

Questo documento presuppone una buona padronanza di Scrum nonché dei termini e del lessico ad esso correlati. Potrebbe essere utile leggere la Guida a Scrum 2020 prima di leggere questo documento. I riferimenti sono inclusi a scopo di attribuzione. Inoltre, la documentazione SGEP correlata, con i relativi riferimenti, fornisce al lettore l'opportunità di esplorare, ricercare e apprendere per una comprensione più ampia e profonda.

I professionisti e gli stakeholders dovrebbero adottare Scrum quando appropriato, con autonomia, senso di urgenza, coraggio, trasparenza, ispezione, adattamento, ritmo, e resilienza. Tutti i soggetti coinvolti dovrebbero puntare a migliorare continuamente per supportare gli obiettivi del prodotto (valore per gli utenti) e dell'organizzazione (impatto per il business). È auspicabile che l'adozione di Scrum vada oltre le linee guida presentate in questo documento e che, così facendo, ispiri una curiosità duratura a esplorare, mettersi in discussione e migliorarsi continuamente.

Questo Pacchetto di Espansione è pensato per supportare molti aspetti della delivery del prodotto da parte di un team auto-organizzato [11], guidato dai bisogni, necessità o desiderata degli stakeholder in risposta a un problema o un'opportunità. Ciò include (ma non si limita a): l'esplorazione del prodotto, lo sviluppo, il rilascio, con conseguente generazione di Valore. Nonostante affondi le sue radici nello sviluppo di prodotti software, Scrum è stato ampiamente adottato in diversi ambiti, permettendo la creazione di Valore attraverso attività di natura complessa [12-17]. Con l'espansione della sua adozione, professionisti come ingegneri, programmati, ricercatori, analisti, avvocati, marketer e scienziati applicano con successo Scrum nei rispettivi settori.

Il termine Valore per gli stakeholder (la SG2020 tratta il concetto di valore solo in termini generici) si riferisce a qualsiasi necessità percepita che uno stakeholder (inclusi, ma non limitati a clienti, figure decisionali e utenti) ritiene importante e che un team rilascia. Tuttavia, gli stakeholder potrebbero non essere sempre consapevoli di ciò che potrebbe essere di valore per loro. L'osservazione o le evidenze potrebbero, intenzionalmente o meno, far emergere valore e influenzare le priorità. Man mano che emergono nuove informazioni, gli elementi potenzialmente di valore dovranno essere identificati, ispezionati, affinati e adattati. Il Valore resta un'ipotesi finché non viene confermato da evidenze, come osservazioni o risultati misurati (e, se considerati nel loro insieme, il loro impatto sull'organizzazione).

Scrum in sintesi

Scrum è un framework per il rilascio di Prodotti complessi [12-17], dove l'esperienza è importante ma di per sé non sufficiente, e il legame causa-effetto risulta chiaro solo

a posteriori. Scrum gestisce l'intero ciclo di vita del Prodotto, che comprende (ma non si limita a): creazione, sostituzione, mantenimento, adattamento, cambiamento continuo, manutenzione e ritiro di funzionalità o dell'intero Prodotto [N.d.T.: ad esempio, quando giunto a fine vita, N.d.T.]. Scrum supporta individui, team e organizzazioni nel diventare e rimanere agili, generando Valore attraverso l'adattamento continuo al cambiamento.

Scrum favorisce un ambiente che facilita la comprensione e la risposta efficace alle necessità degli Stakeholder. Il suo approccio iterativo e incrementale riduce il rischio e promuove il miglioramento continuo. Scrum supporta il team nel trovare un equilibrio tra l'esplorazione dei problemi, l'identificazione delle esigenze degli Stakeholder (inclusi ma non limitati ai clienti), il rilascio di soluzioni, la gestione proattiva del rischio, la validazione del Valore (outcome per gli utenti) e l'osservazione dell'impatto (sui risultati di business)..

Per rischio si intende un qualsiasi fattore che potrebbe provocare un effetto negativo in futuro. Poiché l'esposizione al rischio rimane imprevedibile anche col passare del tempo, anticiparlo è essenziale. L'esposizione al rischio può includere (ma si limita a): rischio di Mercato, adattamento problema-soluzione, adattamento Prodotto-Mercato, tecnologia, rilevamento di sintomi o segnali, reattività, conformità [N.d.T.: es. a normative], azioni correttive, poche opzioni per un compromesso, ecc. Scrum supporta la gestione proattiva dei rischi e favorisce l'individuazione di opportunità.

Scrum promuove la riduzione della distanza tipicamente esistente tra gli Stakeholder - che identificano problemi e/o opportunità - e coloro che lavorano per risolverli o coglierle.

In sintesi, Scrum è basato sull'esistenza di un ambiente nel quale:

1. Gli Stakeholder di supporto, da ora chiamati Supporter, svolgono le attività necessarie per favorire e migliorare in modo proattivo l'adozione di Scrum, sotto la guida e il supporto dello Scrum Master. (La SG2020 menziona gli stakeholder 14 volte, ma non ne fornisce mai una definizione)
2. Un Product Owner definisce il Product Goal, che guida il team nel generare Valore per gli Stakeholder.
3. Il Team Scrum auto-organizzato [11] definisce, perfeziona e trasforma le unità di lavoro selezionate in risultati di valore.
4. Il Team Scrum e gli Stakeholder ispezionano i risultati durante lo Sprint e applicano adattamenti continuui.
5. I Supporter contribuiscono al successo del Team Scrum (Il Supporter non è definito come concetto nella SG2020).
6. *Si ripete il ciclo.*

Per release si intende il processo di creazione di una versione nuova o aggiornata di un Prodotto resa disponibile agli Stakeholder (inclusi, ma non limitati a clienti, decisori e utenti finali). Rappresenta un punto di svolta nel ciclo di sviluppo e determina la transizione del Prodotto dalla fase di realizzazione a quella di utilizzo effettivo e di potenziale generazione di Valore per gli Stakeholder.

Scrum è volutamente incompleto. Anziché prescrivere processi dettagliati, Scrum offre un framework che guida le relazioni e le interazioni con un obiettivo preciso. Scrum permette l'integrazione di diversi processi, tecniche e metodi, ma la loro applicazione varia in base al contesto e alle diverse modalità di utilizzo di Scrum.

Scrum si integra bene con le pratiche esistenti o, in alcuni casi, le sostituisce rendendole ridondanti o addirittura obsolete. Scrum favorisce il miglioramento continuo grazie a una valutazione trasparente dell'efficacia di tutti gli elementi in gioco: i Team Scrum, i Supporter, il management, l'ambiente di lavoro e le tecniche adottate.

Nel contesto del lavoro intellettuale, il termine Scrum, preso in prestito dal gioco del Rugby, è stato coniato da Hirotaka Takeuchi e Ikujiro Nonaka [18] per descrivere team che operavano con approcci tipici delle mischie del Rugby, e dove la conoscenza si diffondeva rapidamente all'interno dell'organizzazione al fine di consegnare Prodotti di assoluta eccellenza.

Teoria a supporto e complementare

Scrum si fonda su un Team Scrum auto-organizzato [11], sull'emergere di pratiche e soluzioni, sull'empirismo [19] e sul Lean Thinking [20]. Si fonda anche su varie teorie a supporto e complementari riportate più avanti nel documento e su idee quali:

- Responsabilità (Accountability),
- Riduzione di tutti quegli sprechi che non apportano Valore (incluse le inefficienze organizzative),
- Approccio al lavoro sia come problemi da risolvere che opportunità da valorizzare,
- Scoperta, creazione e realizzazione di Valore,
- Miglioramento continuo.

Complessità – Il caso in favore di Scrum

In ambiti di lavoro ad alta complessità, come la realizzazione di Prodotti, le incognite superano le certezze, la competenza è preziosa ma da sola non basta e il nesso causa-effetto emerge chiaramente solo a posteriori. Il pensiero della complessità [12-17] offre strumenti e idee di valore che favoriscono nuove intuizioni. I membri del Team Scrum hanno bisogno di tempo per riflettere, aiutarsi a vicenda, correggere il lavoro o cambiare direzione. La diversità cognitiva e l'approccio empirico possono aiutare ad affrontare il lavoro complesso.

Tutto ciò che si ritiene ‘conosciuto’, incluso il Mercato e gli Stakeholder (tra cui, ma non solo, i clienti), potrebbe essere sbagliato. Col passare del tempo alcune aspettative, esigenze o desideri emergono o svaniscono in termini di importanza o urgenza relativa. Un approccio empirico offre strumenti per mettere alla prova queste assunzioni, ispezionare i risultati e adattarsi di conseguenza.

Generalmente, in un determinato contesto nulla resta immutabile, ma cambia e si evolve continuamente. [N.d.T.: si fa frequente riferimento ai 4 tipi di sistema analizzati nel

Framework, citato più avanti: *semplice, ordinato, complesso o caotico*] Il Team Scrum può trovarsi sul bordo del *caos*, dovendo affrontare sfide inedite e lavori mai sperimentati prima. Col tempo, man mano che vengono scoperti schemi ripetitivi (pattern) e applicati approcci euristici, le situazioni diventano meno *caotiche* fino a rientrare in un sistema di tipo *complesso*. Successivamente, in base alle circostanze, il Team Scrum potrebbe trovarsi ad agire in un sistema di tipo *ordinato*, affrontando attività ancora di tipo non *semplice* “ma che rende il lavoro pianificabile”. Tuttavia le cose possono andare anche in senso opposto. È buona pratica per il Team Scrum fermarsi a riflettere e verificare se si trova davvero nel sistema descritto in Cynefin® in base alla realtà attuale. Il punto chiave è che lo sviluppo di Prodotto si scontra spesso con l'imprevedibilità, e Scrum può essere un approccio più efficace rispetto a quelle metodologie che si basano su illusioni di prevedibilità.

Le opportunità che emergono dall’Ispezione e dall’Adattamento di *chi, perché, cosa, come, dove e quando* sono davvero molteplici. È importante attenuare ciò che non funziona e amplificare ciò che funziona. Trasparenza, Ispezione e Adattamento orientati agli obiettivi stabiliti, guidati dal feedback sui risultati (incluse le conseguenze non intenzionali), favoriscono la creazione di Valore, determinano intuizioni, evidenziano rischi e mettono in discussione le assunzioni, promuovendo così il miglioramento continuo.

Costruite fiducia attraverso un team auto-organizzato, attivando Ispezione e Adattamento, assicurando l’erogazione di lavoro di valore e lasciando emergere nuove intuizioni.

Emersione

Per Emersione [21], nel contesto di Scrum, s’intende “processo che determina o permette l’emersione di qualcosa” [N.d.T.], come nel caso di schemi o comportamenti che emergono dalle interazioni all’interno di sistemi *complessi* descritti da Cynefin® [12-17], che non sono predittivi basati sull’osservazione delle singole parti. In Scrum, l’emersione non è strettamente controllata ma è guidata da vincoli abilitanti - come i timebox[N.d.T.: slot di tempo di durata fissa, N.d.T.], i ruoli (reintrodotti in SGEP) e i cicli di feedback - che creano le condizioni per l’auto-organizzazione e l’adattabilità, pur senza imporre risultati precisi. Queste strutture agiscono come ‘isole’ in un mare di imprevedibilità, in modo simile a come in Fisica è possibile che schemi organizzati possano formarsi spontaneamente anche in mezzo al caos, come descritto nel lavoro di Stephen Wolfram [22]. La chiave è che la struttura in Scrum fornisce concetti guida sufficienti da permettere ai team di auto-organizzarsi e, al contempo, far emergere nuove soluzioni, invece di prescrivere il lavoro nel dettaglio.

I Team Scrum, operando come sistemi adattivi complessi, non sono diretti ma vengono influenzati da esperimenti brevi, paralleli e a basso rischio, nonché dal feedback continuo. Schemi e approcci [23] come lo Swarming [N.d.T.: tecnica collaborativa in cui tutti i membri disponibili del team si concentrano insieme su un unico elemento di lavoro - ad esempio una User Story, un bug critico o una funzionalità - finché non è completato], mantenere team stabili e il miglioramento continuo (Kaizen) aiutano a identificare e modellare comportamenti emergenti. Piuttosto che forzare il raggiungimento di risultati, Scrum abilita il Team a scoprire schemi più desiderabili, inclusi - ma non limitati a

- soluzioni innovative o nuovi modi di lavorare, amplificandoli, smorzando quelli non utili.

Questo approccio riconosce che l'auto-organizzazione [11] non è qualcosa che si possa progettare e imporre dall'alto (top-down), ma qualcosa da scoprire in uno spazio favorevole: un ambiente coerente e vitale, in cui si percepisce uno scopo, in cui riecheggia la ‘Quality Without a Name’ di Christopher Alexander [24]. In definitiva, Scrum non tratta l'emersione come un rischio da eliminare, ma come una forza propulsiva e positiva da coltivare per nutrire l'eccellenza nello sviluppo del Prodotto.

Team Scrum Auto-Organizzato

Un Team Scrum auto-organizzato [11] verifica se è sulla giusta strada, interviene quando non lo è, decide in autonomia come lavorare, gestisce i conflitti interni, e affronta i problemi che emergono all'interno del team. Questo implica che, in generale, i manager [25], se presenti nel contesto, non danno istruzioni al Team Scrum su cosa fare, né decidono, direttamente o indirettamente, chi dovrebbe essere “messo da parte” per risolvere determinati problemi. Se i manager sono presenti, è in genere preferibile che esercitino un ruolo di leadership, non di controllo.

I Team Scrum auto-organizzati, focalizzati sul Valore, sono essenziali per la risoluzione creativa dei problemi e per favorire l'emergere di nuove soluzioni. Affidarsi a Team Scrum non auto-organizzati limita la capacità di affrontare la complessità [12-17]. I Team Scrum auto-organizzati [11] non vanno confusi con l'auto-organizzazione individuale. È l'interazione fluida tra i membri a permettere al team di emergere. Favorire l'autonomia del team e la capacità di prendere decisioni più rapide all'interno di una struttura non gerarchica può aiutare i Team Scrum a migliorare la propria auto-organizzazione.

Professionalità

Professionalità significa puntare all'eccellenza e lavorare assieme per rilasciare Valore in modo rispettoso, trasparente e responsabile. Essere professionali significa fare sempre determinate cose e non farne mai altre, indipendentemente dalle circostanze.

Essere professionali significa assumersi la piena responsabilità del Prodotto, dalla sua nascita fino alla dismissione, durante tutto il suo ciclo di vita. Essere professionali significa occuparsi anche della manutenzione - spesso sotto forma di attività operative - che rappresenta un'ottima occasione per i Product Developer di apprendere dai feedback sui risultati delle scelte effettuate a livello ingegneristico. Al riguardo, spesso viene usato il termine DevOps.

Nel contesto dello sviluppo software, la professionalità include, ma non si limita, all'eccellenza tecnica [26]. L'eccellenza tecnica include, ma non si limita, a pratiche come: “Definizione dei requisiti attraverso esempi”, Clean Code, Test Unitari, Test-Driven Development, Test Automatici, Continuous Integration, Continuous Delivery, Architettura e Design, Test di Accettazione, e un approccio determinato e consapevole al testing.

Lean Thinking

“L’approccio Lean Thinking [20] riduce gli sprechi nel lavoro e nei processi, focalizzandosi sul flusso di Valore e sul miglioramento continuo”. I principi Lean si fondano sul miglioramento continuo e sul rispetto delle persone. Adottando i principi Lean, le organizzazioni possono migliorare l’efficacia con costi più bassi nel lungo periodo, offrendo ai Clienti il Valore migliore e favorendo al contempo un ambiente di apprendimento e sviluppo continuo.

Empirismo

L’empirismo [19] è il principio di prendere decisioni basate su prove oggettive o su evidenze osservabili raccolte attraverso cicli di apprendimento, spesso esplorativi. Può essere utile quando l’esperienza da sola non è sufficiente. Scrum si fonda sull’empirismo. Le decisioni si basano su dati concreti o su osservazioni dirette. Un approccio empirico comporta osservazioni continue, sviluppo e refinement delle teorie, la loro messa in pratica e la verifica continua, al fine di creare cicli di feedback efficaci.

L’empirismo può supportare i Team Scrum nel rilasciare ciò che gli Stakeholder considerano di valore quando il *cosa* o il *come* sono incerti. Scrum consiste nel rendere probabile ciò che sembra improbabile attraverso la scoperta, la creazione e la generazione di Valore; questo spesso include, ma non si limita a, compromessi o sperimentazioni. Gli esperimenti sono solitamente fondati su ipotesi verificabili, ma a volte su intuizioni fondate. Una risposta fondamentale alla sperimentazione è il processo decisionale guidato dalle evidenze.

Fermarsi e riflettere combinano elementi di empirismo e lean thinking, creano le basi per trasparenza, ispezione e adattamento verso il Product Goal, e aiutano il Team Scrum e i Supporter a migliorare se stessi e il loro ambiente.

Un’adozione efficace di Scrum riduce la distanza tra gli Stakeholder che presentano i problemi o le opportunità e le persone che li affrontano mantenendo gli obiettivi tangibili e significativi e rilasciando Valore in modo veloce e frequente. Spesso gli Stakeholder hanno un senso errato di certezza sul cosa e sul come. Il Team Scrum spesso ha un senso errato di certezza su chi è impattato. Ispezionare e adattare dovrebbe essere più importante che mantenere promesse o servire gli Stakeholder sbagliati. Tutte le assunzioni possono essere sbagliate. (Il PO nella SG2020 potrebbe essere interpretato come un delegato o un proxy e il valore dato per scontato)

Cadenza

Lavorare in Sprint fornisce un ritmo consistente che aiuta il Team Scrum a concentrarsi su obiettivi chiari e di breve termine. Questa cadenza supporta ispezione e adattamento regolari, abilitando il Team Scrum ad apprendere e adattarsi basandosi sui feedback. Nel tempo, costruisce un ritmo di rilascio sostenibile, migliorando la predicitività e favorendo il miglioramento continuo.

I Tre Pilastri del Controllo del Processo Empirico in Scrum {#the-three-pillars-of-scrum's-empirical-process-control}

L'empirismo, nella sua essenza, è la filosofia secondo cui la conoscenza deriva dall'esperienza e dall'osservazione. Intuizioni di valore emergono da curiosità, esperienza, sperimentazione, dati, visualizzazione e osservazione. Il controllo del processo empirico [27-29] è un metodo per gestire processi complessi [12-17], come quelli in Scrum, adattandosi sulla base dei risultati osservati e che si fonda sui tre pilastri di trasparenza, ispezione e adattamento.

Trasparenza

La trasparenza è un pilastro di Scrum. Mostra la realtà e la chiarezza del lavoro, e rende possibile l'empirismo. La trasparenza permette una percezione più accurata della realtà ed è il punto di partenza per l'Ispezione e l'Adattamento. Il processo emergente, il lavoro e i risultati devono essere visibili a chi esegue il lavoro o riceve gli input sotto forma di obiettivi, elementi del Product Backlog e gli output associati nella forma di Incrementi.

Le decisioni importanti si basano sugli artefatti, sugli esperimenti, sui rilasci o sul feedback sui risultati. Una bassa trasparenza può compromettere l'Ispezione, portando a decisioni che diminuiscono il Valore e aumentano il rischio. La trasparenza abilita l'Ispezione.

Il feedback sui risultati è un dato, idealmente sia quantitativo sia qualitativo, che potrebbe derivare da cambiamenti al Prodotto o dall'ambiente. Contribuisce al Valore per gli Stakeholder, allo sforzo, alle risorse o ai costi. Le persone non sono risorse.

Raggiungere la trasparenza è irrealistico e potenzialmente inapplicabile se esistono inefficienze istituzionali o c'è mancanza di fiducia. Come corollario, Scrum può rendere trasparenti le inefficienze istituzionali e, con volontà collettiva, la fiducia può essere costruita.

Ispezione

L'Ispezione è un pilastro di Scrum. Ispezionare significa osservare la realtà, considerando la direzione del Prodotto (il Product Goal) e l'efficacia del Team Scrum e degli Stakeholder. L'ispezione abilita l'Adattamento. L'Ispezione consiste nel guardare intenzionalmente la realtà e si basa su elementi resi trasparenti, includendo l'evidenza o l'osservazione. Per favorire l'Ispezione e l'Adattamento, Scrum fornisce una cadenza attraverso i suoi eventi.

Gli artefatti di Scrum, gli impegni associati e i progressi verso gli obiettivi concordati devono essere ispezionati frequentemente e con diligenza per rilevare l'emersione [21]. L'ispezione degli artefatti, degli esperimenti, dei rilasci, del mercato o dei feedback sui risultati può produrre apprendimenti o effetti collaterali. Gli effetti collaterali sono risultati o conseguenze inaspettate o non intenzionali.

L'ispezione senza Trasparenza è mal informata, fuorviante e inefficiente.

Adattamento

L'Adattamento è un pilastro di Scrum. Man mano che il Prodotto evolve, ci si aspetta che il Team Scrum e gli Stakeholder si adattino alla realtà non appena emergono opportunità di miglioramento, che si manifestino come risultati di esperimenti, nuove intuizioni, rischi o opportunità. L'Adattamento diventa più difficile quando esistono inefficienze istituzionali o quando le persone coinvolte non sono pronte, disposte o capaci di fare ciò che è necessario.

L'Adattamento inizia con l'accettazione della ‘realità’, basata sull’evidenza. L'Adattamento avviene tipicamente negli Artefatti Scrum, negli impegni associati, nel Team Scrum, negli Stakeholder, nei leader e nell’organizzazione. Se qualsiasi aspetto si discosta da limiti o da confini accettabili, o se il Prodotto risultante è inaccettabile, devono essere apportate modifiche il prima possibile per correggere la rotta.

Senza Adattamento, Trasparenza e Ispezione perdono significato.

I Valori di Scrum

I Valori di Scrum — *focus, apertura* (openness), *impegno* (commitment), *coraggio* e **rispetto** — aiutano a creare un ambiente all’interno del Team Scrum che favorisce la sicurezza psicologica e la collaborazione positiva, che sono in linea con i principi individuati dalle neuroscienze come utili per l’apprendimento e un efficace lavoro di squadra. Tenere presente il contesto è fondamentale.

I Valori di Scrum promuovono la trasparenza e la fiducia, assicurando coerenza tra parole e azioni. Insieme, sono essenziali per creare le fondamenta sulle quali costruire e mantenere collaborazione, prestazioni e coerenza all’interno di un Team Scrum.

L’adozione efficace di Scrum dipende dalla capacità del Team Scrum e dei Supporter (e di altri Stakeholder) di dare l’esempio dimostrando la propria professionalità. I Valori di Scrum possono contribuire a migliorare la fiducia tra il Team Scrum e gli Stakeholder. Inoltre promuovono l’etica [30], il linguaggio condiviso, toni e atteggiamenti adeguati, la modalità di lavoro, i comportamenti e le azioni che favoriscono la costruzione della fiducia. Aiutano anche a colmare — o evitare — il divario che a volte si percepisce tra parole e azioni.

I membri del Team Scrum insieme al nuovo ruolo del Supporter (vedi paragrafo precedente “Scrum in sintesi”, N.d.T.), si impegnano a essere *aperti* riguardo a tutto il lavoro e alle sfide da affrontare. L’umiltà alimenta l’*Apertura*. L’*Apertura* richiede fiducia, e la fiducia, a sua volta, richiede *Apertura*. Il Team Scrum e i Supporter dovrebbero chiedere e offrire feedback costruttivi. Tutte le persone coinvolte si impegnano a collaborare con continuità e ad apprendere continuamente tramite conversazioni ad alto contenuto informativo, supportati da feedback costruttivi sia di tipo qualitativo che quantitativo.

Le conversazioni ad alto contenuto informativo sono interazioni che permettono uno scambio di informazioni più ricco, più rapido e più chiaro. Tipicamente questi dialoghi avvengono in presenza (faccia a faccia), tramite videochiamate, tramite strumenti di

gestione visiva o lavagne (fisiche o digitali) tipo Kanban board su muri, vetrate o attraverso versioni software online, strumenti di collaborazione nella stessa stanza oppure online, situazioni nelle quali i partecipanti possono utilizzare non solo le parole, ma anche il tono della voce, le espressioni facciali, i disegni o il linguaggio del corpo per comprendersi appieno l'un l'altro.

Dato che gli Sprint sono brevi, eventuali fallimenti dovrebbero essere piccoli e rapidi e il rischio viene individuato e gestito grazie a un feedback tempestivo e *aperto*. Forse, l'unico vero fallimento è proprio la mancanza di apprendimento.

Il Team Scrum e i Sostenitori devono avere il *Coraggio* di fare la cosa giusta e affrontare le sfide più difficili. Devono mostrarsi coraggiosi nell'esplorare l'ignoto, nel cambiare direzione, nel chiedere e condividere informazioni e nel partecipare a confronti rispettosi, come conflitti sani e dissensi costruttivi. Il Team Scrum dovrebbe rivolgersi ai Sostenitori e ai leader per chiedere supporto quando necessario.

Il Team Scrum si *Impegna* a raggiungere lo Sprint Goal, effettuando ispezioni e adattamenti verso il Product Goal (aspetto poco evidenziato nella SG2020) e a supportarsi reciprocamente. Impegno significa portare a termine, entro la fine dello Sprint, preferibilmente in anticipo, il lavoro rilevante necessario a raggiungere lo Sprint Goal, rispettando la Definition of Output Done. Inoltre, per Impegno si intende anche raggiungere i risultati attesi attraverso la realizzazione del Valore.

Per i componenti del Team Scrum il principale *Focus* è fare i migliori progressi possibili verso lo Sprint Goal. In secondo luogo, il Focus è rivolto a fare i migliori progressi possibili verso il Product Goal. I Sostenitori si *Impegnano* a garantire uno spazio e un ambiente psicologicamente sicuri affinché il Team Scrum possa consegnare gli Incrementi; all'interno di questo *Focus*, il Team Scrum e i Sostenitori si *Impegnano* a dedicare tempo all'apprendimento continuo e all'Adattamento, e al trasferimento delle conoscenze tra i diversi Team Scrum, per assicurare un'efficacia duratura e consistente nel tempo. Il Team Scrum e gli Stakeholder devono affrontare con consapevolezza i compromessi, valutando sia i successi a breve termine che le conseguenze nel lungo periodo.

Il Team Scrum e i Sostenitori (e gli altri Stakeholder) si *rispettano* reciprocamente come professionisti; *rispettano* le competenze e i punti di vista differenti degli altri e sono costruttivi anche quando non sono d'accordo. Un comportamento rispettoso alimenta la fiducia. Il Team Scrum e i Sostenitori dovrebbero criticare l'idea o l'approccio per individuare opzioni più efficaci, non la persona (o le persone).

Il *Rispetto* aiuta a proteggere dalla strumentalizzazione degli altri Valori di Scrum. Dimostrazioni di *Rispetto* possono includere (ma non sono limitate a): riconoscimento genuino dei successi, supporto reciproco, umiltà, sicurezza psicologica, disaccordo costruttivo e diversità di pensiero.

I membri del Team Scrum e gli Stakeholder possono guardare ai Valori di Scrum facendo riferimento al modello OODA di John Boyd [99, 100, 102]. L'OODA è stato creato dal colonnello dell'Air Force degli Stati Uniti John Boyd per aiutare i piloti a prendere decisioni rapide e intelligenti in situazioni in rapido cambiamento, segu-

do quattro fasi: Osservare (Observe), Orientarsi (Orient), Decidere (Decide) e Agire (Act). È un approccio semplice, continuo, iterativo e potente—spesso utilizzato in modo inconscio—per affrontare l’incertezza come, ad esempio, notare i cambiamenti di mercato (Osservare), analizzare tendenze e rischi (Orientarsi), scegliere le funzionalità di Prodotto da testare (Decidere), e rilasciarle (Agire). L’OODA aiuta le persone a restare flessibili e a rispondere efficacemente a ciò che accade. Scrum può potenziare l’efficacia dell’OODA.

I singoli membri del Team Scrum possono interpretare i Valori di Scrum attraverso il modello OODA di John Boyd, utilizzando Scrum per favorire soluzioni emergenti. Nel contesto di Scrum, i Valori si applicano a tutte le fasi dell’OODA (SGEP pone un’enfasi diversa) e, in particolare, offrono supporto nei seguenti modi:

- Osservare (Observe) — Apertura e Rispetto favoriscono la raccolta di tutte le evidenze rilevanti e di punti di vista differenti.
- Orientarsi (Orient)—Serve Coraggio per interpretare la realtà, affrontare l’incertezza e accettare la necessità di adattarsi o cambiare direzione, potenzialmente sfruttando una pausa riflessiva per mettere in discussione le proprie ipotesi e stimolare nuove intuizioni (insight).
- Decidere (Decide)—Decidere cosa fare richiede un’analisi tempestiva, come l’attività di refinement del backlog, portando i possibili prossimi passi al centro dell’attenzione (Focus) attraverso esperimenti paralleli a basso rischio (safe-to-fail) per testare ipotesi, come sonde (probes), che dovrebbero essere di scala ridotta, condotte in parallelo e progettate in modo che un eventuale fallimento sia gestibile e porti apprendimento.
- Agire (Act)—Quando è chiaro cosa deve essere fatto, perché e da chi, l’Impegno (Commitment) guida il team nell’esecuzione efficace delle attività all’interno di vincoli abilitanti come gli sprint a tempo fisso, creando le condizioni per far emergere soluzioni.

Ulteriori teorie a supporto e complementari

Product Thinking

Le persone traggono beneficio dai Prodotti (inclusi i servizi), non progetti. Un Prodotto è lo strumento attraverso cui si genera Valore, trovando un equilibrio tra benefici a breve e a lungo termine. È per questo che in Scrum esiste il ruolo di Product Owner e non di Project Owner o Project Manager. I Prodotti si sviluppano e si evolvono nel lungo termine e richiedono cura per tutta la loro esistenza, mentre un progetto è limitato nel tempo (time-box) e spesso, una volta concluso, lascia dietro di sé un Prodotto “orfano”.

Il Product thinking [31-33] affronta la tensione [25] insita nel fatto che i Prodotti, spesso, devono concentrarsi (*Focus*) sulla crescita nel breve termine, ma al tempo stesso richiedono di rispondere a esigenze di lungo periodo. Ad esempio: guadagnare popolarità con i primi utilizzatori (early adopters), superare il divario verso l’adozione di massa (crossing the chasm) [34], espandersi, rilasciare nuove versioni, gestire il cambiamento continuo, massimizzare il valore generato verso il Cliente nel tempo e tenere sotto controllo il Costo Totale di Proprietà (Total Cost of Ownership).

Per superare il divario verso l'adozione di massa (“crossing the chasm”) è necessario un cambio di strategia. Si passa dal puntare su Clienti esperti e propensi al rischio, al conquistare utenti, decisori (decision-maker) e Stakeholder più pragmatici e avversi al rischio, focalizzandosi su un mercato di nicchia ben definito o offrendo una soluzione completa e affidabile capace di risolvere problemi concreti. Questo passaggio è fondamentale per trasformare un Prodotto da un successo di nicchia a un’adozione su larga scala, poiché segna il passaggio dall’interesse degli early adopters all’attrazione della maggioranza iniziale. Questi ultimi richiedono spesso prove chiare dell’affidabilità del Prodotto e della sua efficacia in un contesto specifico. Concentrandosi su una nicchia e fornendo una soluzione solida, un’azienda può costruire credibilità, generare una base stabile di Clienti di riferimento e consolidare una posizione forte nel Mercato, colmando efficacemente il divario tra gli early adopters e il Mercato di massa.

I Product Owner devono saper gestire con maestria i compromessi tra le esigenze del presente (il *qui* e *ora*) e quelle previste per il futuro (il *là* e *dopo*) [35] attraverso coraggio, umiltà, confronto, collaborazione, un sano conflitto costruttivo, ecc.

Se le persone coinvolte adottano una visione solo nel breve termine, è probabile che si manifestino conseguenze negative nel lungo periodo come debito tecnico, calo della motivazione del Team Scrum, sovraccarico di attività, attenzione alla sola produttività, ecc. Per questo motivo, sarebbe necessario introdurre fattori di mitigazione che favoriscano la sostenibilità nel lungo periodo.

Per debito tecnico si intende il lavoro extra che si accumula – consapevolmente o inconsapevolmente – quando si scelgono scorciatoie nell’implementazione o nella progettazione per consegnare più velocemente. Con il tempo, questo debito rallenta il lavoro del team, proprio come accade con un vero debito – con l’aggravio di interessi – perché rende più difficili e rischiose le modifiche future. Un gruppo di professionisti cerca di ridurre al minimo il debito tecnico e la trascuratezza. Se si decide di contrarre debito tecnico, questa scelta dovrebbe essere resa trasparente e, se possibile, deve essere messo in atto un piano di mitigazione che emergerà e si adatti nel tempo.

Quando di tratta di Prodotti, Scrum supporta la fattibilità, l’usabilità, la desiderabilità, il Valore e la sostenibilità all’interno di confini etici [30] attraverso:

- la progettazione del Prodotto;
- la gestione del Prodotto;
- una considerazione determinata e coesa delle interazioni tra Stakeholder, ricerca, obiettivi, esplorazione, design, consegna e continua generazione di Valore;
- l’Ingegneria del Prodotto, nel caso specifico di Prodotti tecnologici.

Scrum promuove un equilibrio sano tra breve e lungo termine. L’orientamento all’obiettivo rende possibili risultati mettendo un’ enfasi sul Valore e sulla riduzione del rischio. Lo Sprint Goal (*qui* e *ora*) dovrebbe rappresentare un passo verso il Product Goal (*là* e *dopo*), che apre la strada a risultati di lungo termine. Il Product Goal spesso sostiene la strategia e la Visione di Prodotto.

Il Pensiero Sistemico

Il Pensiero Sistemico (Systems Thinking) [36] riconosce l’interconnessione tra gli elementi nei contesti organizzativi e sociali, comprendendo che le azioni intraprese in una parte del sistema possono generare effetti a catena che non sono sempre prevedibili né lineari nelle altre aree dello stesso sistema. Esperimenti guidati su teorie, cicli di feedback e analisi dei dati a posteriori (follow-up) permettono di far emergere intuizioni di valore e attuabili. Il Pensiero Sistemico offre strumenti e idee di valore e facilita le intuizioni.

Per diventare adattiva [37], un’organizzazione deve evitare le sub-ottimizzazioni locali, come ad esempio ridurre i costi unitari generando un aumento dei costi nel lungo periodo, compromettere gli obiettivi di qualità finendo per perdere la fiducia dei clienti, oppure migliorare un Team Scrum, un flusso o un processo di lavoro che non dovrebbero esistere. In contesti di lavoro complessi [12-17], non è sempre possibile collegare causa ed effetto, tranne che con il senno di poi. Tuttavia, è utile considerare sia gli effetti potenziali sia quelli reali che un intervento può generare a monte, trasversalmente o a valle.

Discovery

La fase di Esplorazione (Discovery) [38-39] inizia spesso con la comprensione delle aspettative, delle necessità e dei desideri delle persone, attraverso osservazione, analisi, conversazioni e sintesi orientata a un risultato desiderato. Una volta raccolte le intuizioni (insight), il Team Scrum inquadra il problema o l’opportunità e li ordina in base al Valore potenziale. Il Team Scrum propone collettivamente possibili soluzioni, senza giudicarle troppo in fretta. Se il Valore potenziale è elevato ma manca evidenza che possa essere realizzato, il Team Scrum dovrebbe condurre ricerche, testare ipotesi o costruire semplici prototipi da verificare con clienti, figure decisionali o utenti reali. La fase di scoperta del Prodotto non si conclude mai; è utile prevedere interviste o osservazioni regolari di clienti, figure decisionali o utenti reali.

La Discovery consiste nell’apprendere con l’obiettivo di raggiungere un risultato desiderato, attraverso la prioritizzazione, l’agire, l’evitare, o il continuo perfezionamento di idee basate sull’osservazione degli utenti, sul loro feedback e su altri apprendimenti. Questo processo enfatizza la collaborazione, la creatività e il fatto di non temere di fallire e riprovare. La Discovery inquadra il lavoro come problemi o opportunità e supporta il Team Scrum nella creazione, prioritizzazione e verifica di soluzioni che bilanciano i desideri delle persone, la fattibilità tecnica e la sostenibilità economica — il tutto, realizzato divertendosi.

Leadership

La leadership è la capacità di influenzare, guidare e ispirare un gruppo di persone verso il raggiungimento di un obiettivo comune, evitando che si crei demotivazione. La Leadership stimola pensieri, azioni e passione, e favorisce indicazioni strategiche chiare; richiede un approccio intenzionale e consapevole per “Andare a vedere, ascoltare e ca-

pire”, raccogliendo fatti e osservazioni per supportare le decisioni, noto come Genchi Genbutsu [40].

La leadership è un processo sociale dinamico che implica responsabilità, costruzione di relazioni e empowerment. Una leadership efficace porta alla co-creazione di una direzione da seguire, all'allineamento efficiente delle risorse e delle persone necessarie e a un mutuo *Impegno (Commitment)* tra i membri del gruppo.

Scrum promuove una forma particolare di leadership, ovvero una leadership orientata alla resilienza, intesa come un insieme di qualità e non come una posizione gerarchica. Perciò, la leadership deve includere, senza limitarsi a, coltivare un ambiente che favorisce Team Scrum auto-organizzati, promuova chiarezza, fiducia, trasparenza, l'esperienza [21] di una direzione, la soddisfazione sul lavoro, abbracciare l'incertezza [41]. Inoltre, spinge a considerare il valore generato dai fallimenti, l'importanza della raccolta di evidenze per decisioni migliori, la gestione proattiva del rischio e la rimozione delle inefficienze organizzative.

La leadership si manifesta da ogni prospettiva, dovrebbe essere presente a tutti i livelli e promuovere la riflessione nei momenti opportuni. Deve perseguire con determinazione il valore, rimanendo però compassionevole ed etica. Richiede di portare avanti con perseveranza una serie di azioni volte a modificare flussi di lavoro, processi, sistemi e l'ambiente lavorativo; ciò include (ma non si limita a) risorse umane, finanza e gestione dei fornitori. Un leader è dunque chi dimostra concretamente la leadership.

Product Owner e Scrum Master bilanciano leadership, autorità e controllo discreto fornendo intenti chiari, stimolando l'iniziativa e rafforzando la responsabilità (accountability). Guidano piuttosto che esercitare un controllo puntuale delle attività (micro-management), assicurandosi che il Team Scrum comprenda la Visione e gli obiettivi, abbia l'autonomia per agire e rimanga responsabile (accountable) dei risultati. Quando necessario, intervengono in modo deciso, pur rispettando la piena responsabilità e autonomia del Team Scrum. I Product Developer dimostrano leadership con l'orientamento al team auto-organizzato, professionalità e focalizzazione sugli obiettivi; l'auto-organizzazione comporta responsabilità. I Supporter mostrano leadership supportando la rimozione degli impedimenti a breve e lungo termine, migliorando la coerenza dei processi di management con Scrum e favorendo, su richiesta, cambiamenti emergenti verso direzioni efficaci.

First Principles Thinking

Il First Principles Thinking è un metodo di risoluzione dei problemi che consiste nello scomporre una sfida nelle sue verità più fondamentali e ricostruire le soluzioni partendo da questi elementi di base. Invece che affidarsi ad analogie o convenzioni consolidate, questa modalità si basa sulla domanda: “*Cosa sappiamo con certezza?*”, per poi ricostruire la comprensione e le soluzioni da questi elementi di base. Esempi di applicazione includono, ma non si limitano a:

- Incoraggiare il Team Scrum a mantenere il *Focus* sui fattori essenziali di efficacia, adattabilità [37] e tempestività—come l'autonomia, la trasparenza e la capacità di adattamento—anziché seguire ciecamente i processi o imitare pratiche altrui.
- Mettere in discussione ogni ipotesi e ricostruire le soluzioni sulla base di fatti e principi essenziali, favorendo così l'emergere di innovazioni dirompenti.
- Promuovere il pensiero originale, il miglioramento continuo e il *Coraggio* di sfidare lo status quo—liberando la creatività e rendendo possibili risultati trasformativi.

Persone e Cambiamento

L'adozione di Scrum presenta un certo livello di difficoltà che non deve essere sottovallutato. Scrum mette a disposizione principi guida, che sono l'essenza dei suoi elementi fondanti, e propone un approccio che resta ancorato ai principi base.

Scrum non va inteso in termini di adozione di strumenti di lavoro né limitato alla rimozione di impedimenti. In Scrum con impedimento si intende qualsiasi cosa che blocca o rallenta l'avanzamento. È cruciale essere determinati, a volte sembrando persino implacabili, ma dimostrando intenzioni costruttive riguardo a persone, cambiamenti e comunicazione. Il cambiamento tipicamente include lo sviluppo delle persone, la progettazione, i flussi di lavoro, i processi, i sistemi, le attitudini, i comportamenti, il linguaggio, le abitudini e il clima lavorativo. La cultura (di lavoro del Team, N.d.T.) è il risultato che emerge da un'evoluzione continua.

Essere efficaci nell'adozione di Scrum significa perseguire un approccio evolutivo, significa avvalersi di agenti del cambiamento, significa un impegno entusiasta da parte di coloro che ne sono impattati o che vogliono creare impatto. La focalizzazione e l'avanzamento quotidiano nell'adozione di Scrum sono elementi prioritari, e pertanto il lavoro di adozione di Scrum non può essere lasciato alla fine, solo a valle di tutte le altre attività di cui si occupa il team.

Iniziate con un cambiamento emergente [N.d.T.: dall'attuale situazione], puntando, con disciplina e costanza, verso una determinata direzione e impegnatevi poi a rendere il cambiamento così normale da diventare routine quotidiana. L'adozione di Scrum ha una direzione ma non necessariamente una destinazione predefinita. Il cambiamento accade e pertanto non è predicibile. La curiosità abilita un modo di percepire, ascoltare, imparare che agevola nel sapersi adattare in una certa direzione. È importante coltivare le relazioni e comprendere punti di vista differenti e anche ascoltare il non detto e vedere ciò che non sta accadendo. Il cambiamento è un lavoro duro, ma appagante. Idealmente agiscono come catalizzatori del cambiamento, trasmettendo positività concreta e apertura alle possibilità.

I Ruoli Scrum nel Pacchetto di Espansione

Un'accountability è un insieme di responsabilità di cui una persona risponde. Ad esempio, il Product Owner è responsabile (accountable) del valore, lo Scrum Master del-

l'efficacia del team Scrum e i Product Developer della creazione di un Incremento utilizzabile. Non tutti i ruoli prevedono accountability, ad esempio gli stakeholder.

I quattro ruoli di Scrum sono Product Owner, Product Developer, Scrum Master e Stakeholder [N.d.T.: i nomi dei ruoli sono mantenuti in inglese, il Product Owner è letteralmente il “Proprietario del Prodotto”, il responsabile dell’evoluzione del prodotto, il Product Developer è la persona che mette a disposizione la propria competenza per creare il prodotto, lo Scrum Master è il “maestro” del processo Scrum, e lo Stakeholder è qualsiasi persona portatrice di un interesse nel lavoro del team e nell’evoluzione del prodotto]. I quattro ruoli fra loro si danno, si ricompensano e si guadagnano fiducia, e consentono quindi una leadership coerente. Nel Team Scrum abbiamo solo tre responsabilità: il Product Owner, il Product Developer e lo Scrum Master.

Una sola persona può ricoprire più di un ruolo Scrum, ma bisogna fare attenzione, quando se ne assume più di uno, a non travalicare i propri compiti. I ruoli Scrum sono progettati per mantenere in atto un sistema di controllo e di equilibrio.

Un Team Scrum è un gruppo di persone che pratica Scrum, ne indirizza le tre *responsabilità* – ovvero Scrum Master, Product Owner e Product Developer – gestisce problemi o opportunità con gli Stakeholder (ossia clienti, utenti, ma non solo), e rilascia Incrementi di prodotto utili, utilizzabili e potenzialmente di valore, sia dalla prospettiva del Team Scrum che degli Stakeholder, al fine di perseguire l’Obiettivo di Prodotto (Product Goal). Per lavori complessi [12-17], un Team Scrum dovrebbe essere piccolo, cognitivamente diversificato e auto-organizzato. I membri del Team Scrum, spesso assistiti dalla tecnologia, si curano reciprocamente del lavoro di ognuno e imparano a svolgere le mansioni degli altri.

Il Team Scrum dovrebbe essere cross-funzionale, il che significa avere competenze multidisciplinari, sia in termini di competenze tecniche che di dominio di business. In un team Scrum non è definita alcuna gerarchia esplicita. Il Team Scrum deve possedere tutte le competenze e il supporto necessari per:

- Imparare (discover) (ossia scoprire, fare ricerca e progettazione), quando necessario;
- Realizzare (incluse competenze ingegneristiche quando appropriato);
- Validare la realizzazione di valore (e, in aggiunta, l’usabilità, la desiderabilità e la fattibilità entro limiti etici [57]).

Il Team Scrum, in collaborazione con i Supporter, si occupa, tutti insieme, del dominio del problema o dell’opportunità, della Product Discovery, della consegna, della verifica e della qualità intrinseca, del go-to-market e della validazione del valore rispetto al Product Goal. Il Team Scrum punta a ottenere miglioramenti chiari, ed essendo auto-organizzato [11], decide *chi fa cosa, come, quando e dove*.

La validazione del valore è la conferma (o smentita), entro determinati limiti, che i risultati attesi siano stati raggiunti.

Il Team Scrum rilascia uno o più Incrementi a ogni Sprint, si auto-organizza continuamente [11] per individuare e risolvere problemi, si sincronizza costantemente e rilascia

frequentemente. Il Team Scrum è sufficientemente piccolo da rimanere agile, ma abbastanza grande da avere la capacità di completare un lavoro significativo entro uno Sprint. Spesso, i Team Scrum più piccoli comunicano meglio e sono più produttivi.

Scrum si basa su Team Scrum auto-organizzati [11] all'interno di una struttura organizzativa o di Prodotto definita. L'autonomia esiste, ma è delimitata dagli eventi Scrum, dalle responsabilità, dagli artefatti, dagli impegni, dai pilastri, dai valori e dai bisogni organizzativi.

Scrum coinvolge gruppi di persone che insieme possiedono o acquisiscono tutte le competenze necessarie per svolgere il lavoro, condividendo tali competenze quando necessario. Interazioni perseguite con determinazione, supportate dai leader, sono necessarie per migliorare le probabilità di successo.

Il *Focus* deve rimanere sull'ottenimento del Product Goal nel modo più efficace, con il giusto livello di investimento e fornendo risultati di valore.

Scrum promuove il lavoro di squadra collaborativo incoraggiando l'interazione continua e la responsabilità condivisa, anziché un lavoro sequenziale e compartimentato. Questo approccio consente al Team Scrum e agli Stakeholder di abbracciare l'incertezza [41], permettendo aggiustamenti rapidi grazie all'apprendimento e al feedback continuo. La natura sovrapposta della scoperta, dello sviluppo e della validazione del valore garantisce un approccio più adattivo [37] e orientato al valore nello sviluppo del Prodotto.

Sovrapporsi nello svolgimento delle attività favorisce la responsabilità condivisa all'interno del Team Scrum, aumentando il coinvolgimento e l'impegno. Il Team Scrum concentra l'attenzione nell'affrontare sfide e opportunità, incoraggia comportamenti proattivi, coltiva un insieme di competenze diversificate e accresce la consapevolezza delle dinamiche di mercato tra tutti i partecipanti.

Il Team Scrum si occupa di tutte le attività legate al Prodotto, dalla collaborazione con gli Stakeholder alla validazione del valore, includendo verifica, manutenzione, operatività, sperimentazione, ricerca e sviluppo, e tutto ciò che può essere necessario. Il Team Scrum infonde qualità. I Supporter garantiscono che l'organizzazione strutturi il contesto e l'ambiente di lavoro e abiliti il Team Scrum ad auto-organizzarsi [11]. Lavorare per Sprint a un ritmo sostenibile migliora il *Focus* e la coerenza.

Il Team Scrum e gli Stakeholder non possono sapere cosa impareranno. Alcune scoperte forniranno maggiore certezza, altre genereranno ulteriore incertezza [41], alcune cose possono emergere, scomparire, perdere importanza o diventare meno prioritarie.

Un Team Scrum ha autonomia allineata con la strategia. Ciò significa che il Team Scrum ha la libertà di decidere come risolvere i problemi, mantenendo la concentrazione su obiettivi e risultati condivisi, abilitando sia l'innovazione sia la coerenza organizzativa. Favorire la diversità cognitiva nel Team Scrum è essenziale. Contaminarsi reciprocamente è più probabile quando i membri del Team Scrum collaborano, si fidano l'uno dell'altro e sanno riflettere su se stessi.

Per ottenere risultati efficaci, il Team Scrum e i Supporter dovrebbero coltivare la disponibilità a disimparare principi obsoleti. Ispezione e Adattamento richiedono un contesto che anticipi e tolleri gli errori. È essenziale *focalizzare* le critiche sulle idee e non sulle persone. Tutti i membri del Team Scrum “giocano sullo stesso campo”, con un lavoro coerentemente sovrapposto, e sono tutti responsabili del successo.

Stakeholder

Lo Stakeholder è un ruolo. Uno Stakeholder è un’entità, un individuo o un gruppo interessato a, influenzato da o capace di influenzare input, attività e risultati. Gli Stakeholder possono avere un interesse diretto o indiretto all’interno o all’esterno dell’organizzazione, dei suoi Prodotti o servizi.

Esempi di Stakeholder includono (ma non sono limitati a): clienti, figure decisionali, utenti, fornitori, figure influenti, manager, colleghi, leader, legislatori, sponsor finanziari, esperti di dominio e responsabili della governance. Anche Stakeholder inanimati o non umani, come la legge o l’intelligenza artificiale, non vanno ignorati. Alcuni Stakeholder hanno un impatto maggiore o sono più impattati di altri, e ciascuno può dare priorità a fattori diversi. Ogni Stakeholder ha un grado diverso di potere o influenza.

Un cliente è un qualsiasi Stakeholder che riceve valore dal Prodotto che ha scelto e/o acquistato. I clienti possono includere gli acquirenti veri e propri (chi paga o acquisisce il Prodotto), figure decisionali (chi approva o autorizza l’acquisizione) e utenti finali (chi interagisce direttamente con il Prodotto). Talvolta il cliente non coincide con il cliente finale, ad esempio nei modelli B2B2C [42] o B2B2B [43].

Perché l’adozione di Scrum sia efficace, è fondamentale avere interazioni determinate e regolari tra gli Stakeholder (inclusi ma non limitati a clienti e utenti) e il Team Scrum.

Un utente è uno Stakeholder che interagisce direttamente con il Prodotto per raggiungere obiettivi specifici o per risolvere problemi. Gli utenti vivono in prima persona l’utilizzo del Prodotto – che sia un servizio, una piattaforma o un’esperienza – e il loro feedback e livello di soddisfazione sono fondamentali per il miglioramento continuo del Prodotto stesso. Gli utenti potrebbero anche non influenzare la decisione d’acquisto, ma la loro partecipazione e coinvolgimento sono essenziali per il successo continuo del Prodotto. Talvolta, l’utente non coincide con l’utente finale, ad esempio in contesti B2B2C o B2B2B. Perché l’adozione di Scrum sia efficace, è fondamentale avere interazioni determinate e regolari con gli utenti (e, se possibile, con gli utenti finali).

Una figura decisionale (chiamato anche “chooser” da Jeff Patton) [44] è uno Stakeholder con l’autorità di approvare, selezionare o autorizzare l’adozione o l’acquisto del Prodotto; è responsabile della valutazione di diverse opzioni e della scelta finale, e in generale tiene conto dei bisogni sia degli utenti sia dell’organizzazione nel suo insieme. Una tale figura potrebbe anche non usare il Prodotto, ma la sua scelta influenza direttamente quali Prodotti vengono adottati e come il valore è reso disponibile ad altri Stakeholder. Perché l’adozione di Scrum sia efficace, è spesso meglio procedere anche con informazioni imperfette, raccogliendo il feedback che emerge dai risultati.

I legislatori (e, per gli scopi di questo documento, anche i redattori di policy interne o esterne) stabiliscono regole, policy e limiti entro cui opera il Prodotto. Definiscono quadri normativi, regolatori o organizzativi che plasmano la distribuzione del valore del Prodotto agli Stakeholder e stabiliscono standard di professionalità. Garantiscono che il Prodotto sia allineato a vincoli esterni o interni, e ne guidano di conseguenza l’evoluzione e la sostenibilità. Perché l’adozione di Scrum sia efficace, è fondamentale non esagerare né sottovalutare i requisiti legali.

Gli sponsor finanziari forniscono fondi e risorse per lo sviluppo, il lancio e il miglioramento del Prodotto. Valutano la fattibilità, il valore e la sostenibilità del Prodotto, investendo sulla base del suo potenziale di generare valore continuo per gli Stakeholder. Influenzano la visione, la strategia e gli obiettivi del Prodotto per garantire coerenza con il ritorno sull’investimento e la sostenibilità a lungo termine. Perché l’adozione di Scrum sia efficace, è fondamentale mantenere flessibilità mentale e flessibilità finanziaria con l’emergere di nuove informazioni.

Gli esperti di dominio contribuiscono con conoscenze approfondite o competenze uniche, essenziali per la creazione, l’evoluzione e la manutenzione del Prodotto. Che si tratti di tecnologia, design, conformità o specifici ambiti professionali, queste persone supportano l’usabilità, la fattibilità, la professionalità e l’estendibilità del Prodotto, ma non devono ostacolare l’auto-organizzazione del Team Scrum [11]. Possono aiutare a colmare lacune di soddisfazione e contribuire all’adattabilità del Prodotto, aiutando a identificare, rappresentare o misurare ciò che dovesse emergere [21]. Perché l’adozione di Scrum sia efficace, è essenziale promuovere un trasferimento regolare di conoscenze dagli esperti di dominio al Team Scrum e tra i suoi membri.

Il termine “divario di soddisfazione” indica la differenza tra ciò che gli Stakeholder sperimentano oggi e ciò che vorrebbero sperimentare. In altre parole, è la distanza tra il livello attuale di soddisfazione rispetto a quello che vorrebbero raggiungere.

La governance si riferisce a strutture, standard, regolamenti, norme, processi e pratiche che limitano e guidano consapevolmente la direzione, le decisioni e le responsabilità legate al Prodotto. La governance favorisce la trasparenza e guida il rispetto degli standard di valore, sostenibilità e professionalità. Inoltre, fornisce meccanismi per la gestione dei rischi e l’adattamento del Prodotto a esigenze o contesti in evoluzione, e ne supporta la natura duratura ed evolutiva. Perché l’adozione di Scrum sia efficace, è fondamentale che la governance sia coerente con Scrum, ad esempio con un unico punto di contatto per ciascuna area di governance, che abbia interazioni perseguite con determinazione su qualità e conformità con il Team Scrum, con regolari momenti di Ispezione e Adattamento della governance stessa, e nessuna sorpresa.

Supporter

Un Supporter è un tipo specifico di Stakeholder. I Supporter sono, appunto, Stakeholder di supporto e agenti del cambiamento. Spesso fanno parte di un gruppo manageriale influente [45], in grado di ispirare e rimuovere i fattori di demotivazione. I Supporter sostengono il Team Scrum affinché possa prosperare e influenzano i flussi di lavoro, i processi, i sistemi, i Prodotti, i servizi e l’ambiente di lavoro dell’organizzazione affin-

ché siano coerenti con l'adozione di Scrum e con l'emergere del Valore [21]. I Supporter dovrebbero partecipare quando e dove necessario, o su richiesta. La creazione di Valore richiede spesso una collaborazione efficace e costruttiva con altri Stakeholder.

A seconda delle dimensioni dell'organizzazione, esempi di Supporter includono (ma non sono limitati a): colleghi, decisori, sponsor finanziari, responsabili della governance, manager, esperti di dominio, marketing, HR, finanza, ufficio acquisti e early adopters (utilizzatori precoci, pionieri). I Supporter che non abilitano il Team Scrum a fare ciò che è raccomandato in questo documento non sono realmente Supporter. Dirigenti e membri del consiglio di amministrazione hanno un ruolo cruciale nel creare un contesto in cui i Supporter possano davvero supportare. I Supporter dovrebbero esercitare una leadership apprezzata dal Team Scrum.

Intelligenza Artificiale

L'Intelligenza Artificiale (IA) è sempre più integrata nei contesti lavorativi e può potenziare in modo significativo le capacità di un Team Scrum, nella fase di esplorazione (discovery), nel processo decisionale, nello sviluppo del Prodotto e nella creazione di Valore.

L'Intelligenza Artificiale può potenziare Scrum attraverso:

- il controllo empirico del processo [27-29]: le analisi guidate dall'Intelligenza Artificiale migliorano la trasparenza, l'ispezione e l'adattamento.
- la crescita cognitiva: l'Intelligenza Artificiale consente ai membri del Team Scrum di concentrarsi su aspetti strategici, creativi ed etici.
- il continuo adattamento del valore: l'Intelligenza Artificiale può aggiornare e ridefinire le priorità degli elementi del Product Backlog sulla base di feedback degli utenti in tempo reale e le tendenze di mercato.
- la visione sistemica: l'Intelligenza Artificiale identifica interdipendenze nasconde, migliorando il processo decisionale basato sui dati.

I pericoli sono infiniti. È essenziale mantenere una chiara responsabilità umana sui risultati (in linea con le responsabilità definite da Scrum), usando l'Intelligenza Artificiale come un potente partner decisionale, ma supervisionato. Questo approccio è noto come "human in the loop" ("l'essere umano come parte del ciclo"). Sebbene l'Intelligenza Artificiale possa aumentare l'innovazione e l'efficacia a costi contenuti, non sostituisce la responsabilità umana. Infatti, l'Intelligenza Artificiale deve supportare, non sovertire, il controllo empirico del processo [27-29] e il processo decisionale etico [30] di Scrum. Il Team Scrum resta responsabile della consegna di risultati di valore, della valutazione delle evidenze e del mantenimento della professionalità.

Ma anche le possibilità sono infinite. I team Scrum possono avvalersi dell'IA per:

- Migliorare l'attività di Product Backlog Refinement
- Scoprire ambiguità nei documenti e ispezionare continuamente le proprie raccomandazioni e risultati relativamente a preconcetti, errori e conseguenze indesiderate.
- Validare e adattare regolarmente modelli e applicazioni.

- Favorire la trasparenza nell'ordinamento e stabilire le priorità del Product Backlog.
- Creare agenti con ruolo di membri non-umani del team.
- Testare e mettere in discussione in modo intenzionale il pensiero corrente.

L'Intelligenza Artificiale può essere uno strumento di supporto se utilizzato con buone intenzioni. Gli strumenti di Intelligenza Artificiale dovrebbero essere valutati come qualsiasi altro contributore al flusso psicologico [46] e all'apprendimento: questi strumenti migliorano i cicli di feedback? Aiutano a validare più velocemente le ipotesi? Agiscono, quando lo fanno, secondo comportamenti etici [30]?

Il flusso psicologico [46], al pari della la trance agonistica, è uno stato di completa immersione e soddisfazione in un'attività, quando azione e consapevolezza si fondono e il tempo sembra scorrere in modo diverso.

Scrum incoraggia il Team Scrum a sperimentare in modo responsabile con l'Intelligenza Artificiale, attraverso prove a impatto limitato, tipicamente reversibili. Adattamento e ispezione si applicano non solo al Prodotto, ma anche all'integrazione dell'Intelligenza Artificiale nel processo di delivery.

L'attenzione deve comunque rimanere sulle dinamiche umane di collaborazione e lavoro di squadra, con l'Intelligenza Artificiale posizionata come un potenziale strumento di supporto per ottenere adattabilità con rapidità.

Product Developer

Il Product Developer è un ruolo e una responsabilità. Tutti i Product Developer, nel loro insieme, dovrebbero possedere tutte le competenze necessarie per creare gli Incrementi di Prodotto. Questo insieme di competenze viene spesso definito cross-funzionale.

Un Product Developer può essere una persona umana o una macchina. I Product Developer umani sono *Impiegati* nella creazione, ricerca, ispezione e adattamento di ogni aspetto di un Incremento rilasciabile a ogni Sprint. Il loro *Focus* principale è sullo Sprint corrente, anche se una parte del loro tempo può essere dedicata al refinement di attività future e all'analisi dei feedback su risultati, effetti collaterali o nuove conoscenze.

I Product Developer aderiscono alla Definition of Output Done e perseguono un miglioramento netto. Raggiungono i risultati migliori quando si *focalizzano* esclusivamente su un solo Prodotto. Se, in un dato momento, il Product Owner o lo Scrum Master lavorano attivamente su elementi dello Sprint Backlog, lo fanno in qualità di Product Developer.

I Product Developer dovrebbero adottare comportamenti adeguati alla situazione, tra cui (ma non solo): collaboratore, creatore, promotore della qualità tecnica, dell'esplo-razione (discovery), del rilascio (delivery) e della validazione del valore.

Almeno un Product Developer deve essere umano. Più Product Developer umani spesso migliorano la diversità cognitiva, utile per affrontare la complessità.

I Product Developer sono sempre, tutti insieme, responsabili di:

- Creare un piano emergente nello Sprint Backlog per raggiungere lo Sprint Goal;
- Instillare qualità, aderendo e migliorando la Definition of Output Done;
- Creare almeno un Incremento utilizzabile ad ogni Sprint;
- Apprendere tipicamente orientati dai dati determinati dalla Definition of Outcome Done;
- Adattare il piano quotidianamente in funzione dello Sprint Goal;
- Responsabilizzarsi a vicenda come professionisti;
- Perseguire un miglioramento netto.

Il contesto conta ed è fondamentale considerare le circostanze specifiche, sebbene, come regola generale, un Product Developer che non è disposto, pronto o in grado di comportarsi da professionista dovrebbe fare un passo indietro e non ricoprire il ruolo di Product Developer.

Product Owner

Il Product Owner è un ruolo e una responsabilità. Il Product Owner deve essere una persona umana e per essere efficace, deve essere un leader per il Prodotto: deve massimizzarne il valore nel lungo termine, deve sapere come individuarlo e capire quando è necessario. Lavora a tutti i livelli e in tutte le aree di business rilevanti dell'organizzazione, collaborando con Stakeholder, Scrum Master e Product Developer per creare valore.

Il Product Owner usa il Product Backlog per definire cosa verrà realizzato e, approssimativamente, in quale ordine. Ha sempre ben in mente qual è il Product Goal, e il suo Focus principale, in ogni momento, è massimizzare il valore nel lungo periodo.

Il ruolo di Product Owner non è rappresentato da colui che analizza e descrive dettagliatamente gli elementi del backlog. Ogni minuto speso a non fidarsi dei Product Developer è un'occasione persa per pensare in modo più strategico, per lavorare con gli Stakeholder o per creare più valore. Il Product Owner dovrebbe adottare comportamenti adeguati alla situazione, tra cui (ma non solo): essere un visionario, un rappresentante del cliente, un collaboratore, una figura influente, uno sperimentatore, una figura decisionale, un promotore del coinvolgimento degli Stakeholder, della chiarezza, della qualità del Prodotto e della realizzazione del valore.

Il Product Owner è anche responsabile:

- del coinvolgimento degli Stakeholder, del capire il loro potere, le loro aspettative, i loro bisogni e desideri;
- dell'ascolto, apprendimento e adattamento continui;
- del bilanciamento continuo fra diversi compromessi, tra cui:
 - Qualità, velocità, competenza, riduzione del rischio, valore, semplicità [47];
 - Stakeholder con aspettative e limiti spesso in conflitto;
 - Valore, clima di lavoro (la sostenibilità del team Scrum è importante), clienti potenziali;
- dello sviluppo e comunicazione esplicita del Product Goal;

- della creazione e comunicazione efficace di una narrazione coerente del Prodotto;
- della creazione e comunicazione chiara degli elementi del Product Backlog;
- dell’ordinamento degli elementi del Product Backlog;
- della garanzia di trasparenza e comprensione del Product Backlog;
- della comunicazione efficace dei risultati supportati da misurazioni nella Definition of Outcome Done;
- della decisione finale sulla Definition of Outcome Done;
- della promozione della creazione, esplorazione, delivery e validazione del valore;
- di altre attività di Product management, se richieste.

Il Product Owner può svolgere direttamente queste attività o collaborare con il Team Scrum per concordare chi le svolge. In ogni caso, ne rimane responsabile.

Per avere successo, tutti gli Stakeholder e Supporter devono *rispettare* le sue decisioni, che si riflettono nel contenuto e nell’ordinamento del Product Backlog e nell’Incremento ispezionabile durante la Sprint Review. L’autorità al Product Owner gli viene delegata dall’organizzazione.

La Product Ownership richiede forti competenze di Product management e di dominio. Chi ne è privo potrebbe aver bisogno di supporto e guida fino allo sviluppo delle competenze necessarie. Il contesto conta, ma come regola generale, un Product Owner che non è disposto, pronto o in grado di acquisire competenze da Product Manager dovrebbe fare un passo indietro e non ricoprire il ruolo di Product Owner. Un esperto di dominio non è necessariamente la scelta migliore, poiché serve anche leadership e capacità di gestione del Prodotto: in questo caso, ad esempio, una responsabilità da Product Developer potrebbe essere più appropriata.

Se il Team Scrum lavora su più Prodotti, piattaforme o progetti, ciascun manager coinvolto dovrebbe qualificarsi come Stakeholder (e Supporter) del Product Owner, collaborando per massimizzare il valore nel lungo periodo. Scrum incoraggia il Team a rimanere *Focalizzato e Determinato*, aiutandolo a rilasciare risultati di valore ed evitando l’insidia di diventare una “fabbrica di funzionalità”.

Il Product Owner è una sola persona, non è un comitato né una tecnologia. Chi vuole modificare il Product Backlog deve convincere il Product Owner, che massimizza il valore nel lungo termine e spesso, nel fare questo, implementa gli opportuni compromessi.

Scrum Master

Lo Scrum Master è un ruolo e una responsabilità. Lo Scrum Master deve essere una persona umana. È un agente del cambiamento che opera a tutti i livelli dell’organizzazione e in tutte le aree aziendali. Lo Scrum Master guida con l’esempio e favorisce l’efficacia del Product Owner, del Team Scrum, degli Stakeholder e dei Supporter nell’adozione di Scrum. Lo Scrum Master comprende la complessità [12-17] ed è abile nel facilitare il prossimo passo giusto da compiere.

Lo Scrum Master dovrebbe adottare comportamenti appropriati in base alla situazione; questi comportamenti includono (ma non sono limitati a): essere guida, coach, mentore,

insegnante, osservatore, colui che rimuove impedimenti, agente del cambiamento, facilitatore dell'efficacia e promotore del miglioramento continuo. Lo Scrum Master non è un amministratore del team, un manager degli stati di avanzamento, un controllore di attività, un emanatore di regole, uno che prenota sale riunioni, un amministratore di dashboard di reporting, un presidente, un eroe, un coordinatore ma neanche uno Scrum Master “assente” che lascia tutto all’“auto-organizzazione”.

Lo Scrum Master è responsabile dell’efficacia del Team Scrum, degli Stakeholder, dei Supporter e delle persone coinvolte nell’abbracciare l’empirismo [19], l’auto-organizzazione e l’adozione di Scrum come descritto in questo documento. Lo Scrum Master affronta qualsiasi cosa ostacoli o rallenti i progressi del Team Scrum e che non può essere risolta dal team stesso.

Lo Scrum Master supporta il Team Scrum, il Product Owner e i Supporter in diversi modi, tra cui:

- Aiutare tutti a comprendere la teoria e la pratica di Scrum, offrendo formazione o coaching quando necessario;
- Abilitare il Team Scrum e i Supporter a migliorare continuamente in vari ambiti;
- Favorire interazioni tempestive, mirate e determinate;
- Assicurarsi che il Team Scrum abbia una Definition of Output Done adeguata;
- Garantire che tutti gli eventi Scrum avvengano e siano costruttivi, produttivi e si chiudano nel tempo previsto;
- Provocare la rimozione di impedimenti legati al lavoro sul Prodotto e all’adozione efficace di Scrum;
- Fare coaching sull’auto-organizzazione [11] e la cross-funzionalità;
- Aiutare il Team Scrum, gli Stakeholder e i Supporter a comprendere il proprio ruolo nel supportare Incrementi di alto valore che rispettino la Definition of Output Done, in direzione del Product Goal e della Definition of Outcome Done;
- Migliorare l’adattabilità [37] e ottimizzare il flusso di valore;
- Promuovere la fiducia basata sulle evidenze, ma con coinvolgimento emotivo, compassione e tempestività per evitare un’eccessiva sicurezza;
- Coltivare la capacità di operare come agenti del cambiamento, sia per il Team Scrum che per i Supporter;
- Incoraggiare comportamenti utili all’interno del Team Scrum, allineati con i Valori Scrum, per favorire fiducia, collaborazione e alta performance;
- Sostenere il Team Scrum nel consegnare lavoro di valore, ricevere feedback e apportare modifiche rapidamente e frequentemente, quando necessario.

Lo Scrum Master supporta il Team Scrum in diversi modi, tra cui:

- Supportare il Team Scrum nella sua formazione, nella crescita delle competenze e nel miglioramento continuo;
- Aiutare il Team Scrum a comprendere l’importanza di avere elementi del Product Backlog chiari, concisi e orientati al valore;
- Vigilare affinché l’intero Team Scrum collabori in modo determinato e con uno scopo preciso, sia internamente sia con gli Stakeholder, rispettando la Definition

of Output Done e concentrandosi sulla creazione di Incrementi di alto valore in linea con la Definition of Outcome Done.

Lo Scrum Master supporta il Product Owner in diversi modi, tra i quali:

- Aiutare a individuare tecniche per una definizione efficace del Product Goal e per la gestione del Product Backlog;
- Supportare l'adozione di pianificazione di Prodotto di tipo emergente, più adatta ad un contesto Complesso [12-17];
- Aiutare il Product Owner a esprimere gli outcome (risultati attesi) come misurazioni attraverso la Definition of Outcome Done;
- Aiutare il Product Owner a comprendere l'importanza di avere elementi del Product Backlog chiari, concisi e orientati al valore;
- Aiutare il Product Owner a mantenere il Focus sulla realizzazione del valore.

Lo Scrum Master supporta gli Stakeholder in diversi modi, tra cui:

- Quando serve più della pura competenza, aiuta le persone coinvolte e gli Stakeholder a comprendere e adottare:
 - Un approccio empirico per il lavoro complesso [12-17], dove causa ed effetto risultano chiari solo a posteriori;
 - Strategie che vanno oltre il controllo empirico del processo, ad esempio conducendo più esperimenti paralleli “safe-to-fail”, cercando nuove idee, praticando l'exaptation o testando intuizioni fondate. Exaptation significa utilizzare qualcosa creato o impiegato per uno scopo e usarlo in modo diverso, specialmente in situazioni nuove o poco chiare.
- Promuovere azioni che supportino il mantra: “Smetti di iniziare cose; inizia a finirle”;
- Facilitare la collaborazione degli Stakeholder, su richiesta o in base al bisogno;
- Aiutare gli Stakeholder a comprendere l'importanza di avere elementi del Product Backlog chiari, concisi e orientati al valore;
- Aiutare gli Stakeholder a mantenere il *Focus* principalmente sulla realizzazione del valore.

Lo Scrum Master lavora con i Supporter in diversi modi, tra cui:

- Guida, fa formazione e coaching ai Supporter nell'adozione di Scrum;
- Chiarisce cosa ostacola un'adozione efficace di Scrum;
- Facilita un cambiamento disciplinato ed emergente nella direzione di supportare l'adozione di Scrum;
- Promuove cambiamenti organizzativi orientati a facilitare la delivery piuttosto che la sola gestione.

Lo Scrum Master lavora con l'organizzazione in diversi modi, tra cui:

- Guida, fa formazione e coaching all'organizzazione nell'adozione di Scrum;
- Pianifica e fornisce consulenza sull'adozione di Scrum all'interno dell'organizzazione;
- Collabora con i reparti correlati su come possono supportare l'adozione di Scrum;
- Rimuove gli ostacoli all'adozione di Scrum.

Gli Scrum Master possono lavorare in squadra con altri Scrum Master o Supporter per supportare l'intera organizzazione; possono anche collaborare con altri agenti del cambiamento o leader, quando necessario. Lo Scrum Master, in quanto agente del cambiamento, è responsabile della qualità dell'adozione di Scrum e dovrebbe collaborare con altri agenti del cambiamento per migliorarla.

Lo Scrum Master è una persona, non è un comitato né una tecnologia, ed è a servizio del Product Owner, del Team Scrum, degli Stakeholder e dell'intera organizzazione. In quanto agente del cambiamento e leader, lo Scrum Master dovrebbe generalmente invitare le persone a partecipare al cambiamento. È utile che lo Scrum Master comprenda il flusso di valore [48-49], il pensiero lean [20], la teoria della complessità [12-17] e altre teorie di supporto e complementari descritte in questo documento, oltre ad aiutare le persone nel *come fare*. È anche utile che lo Scrum Master sia determinato e abbia un'insaziabile voglia di apprendere e di cambiare.

Essere uno Scrum Master è una vocazione, in cui aiutare gli altri ad avere successo è una ricompensa sufficiente. Uno Scrum Master non cerca i riflettori. Come ogni buon leader, dà credito agli altri e si assume la responsabilità quando le cose vanno male. Rimanere nel ruolo a lungo può aiutare il Team Scrum a raggiungere il suo pieno potenziale, ma solo se i Product Developer sviluppano collettivamente l'auto-organizzazione. Un comportamento da "genitore" da parte dello Scrum Master non favorisce un Team Scrum auto-organizzato. Il contesto è importante, ma, come regola generale, uno Scrum Master che non è disposto, pronto o capace di essere un agente del cambiamento dovrebbe fare un passo indietro dal ruolo.

Gli Artefatti di Scrum nel Pacchetto di Espansione

Gli Artefatti di Scrum forniscono trasparenza su ciò che il Team Scrum e gli Stakeholder ritengono possa generare valore: questo permette che tutti possano riferirsi alla stessa base comune per l'Ispezione e l'Adattamento.

Ogni Artefatto è collegato ad uno specifico Impegno (Commitment):

- Per il Prodotto al servizio degli Stakeholder, è la Definition of Outcome Done (aggiunta in SGEP).
- Per l'Incremento che rappresenta un aggiornamento potenziale del Prodotto, è la Definition of Output Done (rinominata nel SGEP).
- Per il Product Backlog, è il Product Goal.
- Per lo Sprint Backlog, è lo Sprint Goal.

Al rilascio dell'Incremento (considerato come l'Output), ciò che crea Valore (considerato come il Risultato) è il Prodotto in sé. Per Valore si intende qualsiasi creazione misurabile e osservabile o realizzazione di aspettative, necessità o desideri valutati dal punto di vista degli Stakeholder.

Questi impegni rafforzano i tre Pilastri di Scrum, ovvero Trasparenza, Ispezione e Adattamento, consentendo il controllo empirico del processo [27-29]. Il Product Goal rimane fisso finché non emergono evidenze o osservazioni contrarie nella Definition of Outcome Done del Prodotto in oggetto. La Definition of Outcome Done non deve essere

indebolita durante lo Sprint. Quindi, cosa potrebbe essere ridotto o modificato? Potrebbero ad esempio essere i Criteri di Accettazione (Acceptance Criteria) per uno specifico elemento del Product Backlog, oppure l'implementazione o la conformità di una specifica funzionalità, o anche elementi alternativi del Product Backlog per raggiungere lo Sprint Goal. L'elenco potrebbe essere lungo.

Se il Product Goal viene modificato spesso, potrebbe essere il sintomo che qualcosa non va, forse a causa di una mancanza di Focus su ciò che conta. Focus significa essere professionali e decidere su cosa lavorare, ma anche su cosa non lavorare.

Prodotto

Il Prodotto è uno degli artefatti di Scrum, il più importante. Un Prodotto non deve essere necessariamente un oggetto tangibile ma anche un'esperienza olistica o una piattaforma. Può essere anche un servizio, un elemento digitale, fisico o ibrido: ciò che conta, indipendentemente dalla sua natura, è che il Prodotto offre Valore continuo alle parti interessate (inclusi, ma non limitati agli utenti finali).

Per esperienza s'intende una soluzione specifica progettata per soddisfare le esigenze degli Stakeholder, incluso l'utente finale, tipicamente - ma non necessariamente - esterni all'organizzazione. L'esperienza intende fornire un'interazione diretta, che punta a generare Valore. In genere, è focalizzata sulla risoluzione di un particolare problema, lo sfruttamento di una opportunità, o di un insieme di esse pensate e rivolte agli Stakeholder come, a titolo esemplificativo ma non esaustivo, clienti, utenti e chiunque in condizione di prendere decisioni.

Per piattaforma si intende l'infrastruttura di base (inclusi i suoi componenti fisici e logici) e gli elementi architetturali o l'insieme di strumenti di sviluppo, frequentemente di tipo software, che consente agli sviluppatori di creare Prodotti capaci di fornire un'esperienza. Le piattaforme forniscono una base su cui sviluppare più Prodotti, concentrando su scalabilità, affidabilità e flessibilità per gli sviluppatori e/o contributori, piuttosto che sull'interazione diretta con l'utente.

Il Team Scrum e gli Stakeholder devono sempre avere una chiara comprensione di cosa sia il Prodotto, chi siano i clienti, gli utenti o i decisori finali, nonché di che tipo di Prodotto si tratti - ad esempio uno per utenti finali, uno per i dipendenti o per il Team Scrum - che può avere Stakeholder differenti, nonché modi diversi per creare Valore. Un Prodotto è per sua natura evolutivo e spesso di lunga durata. Il Prodotto necessita di un unico Product Backlog per aumentare la trasparenza e massimizzare il valore.

Il contesto è importante, ma come regola generale, affinché un Prodotto crei e mantenga la sua spinta propulsiva, è utile che:

- Indirizzi le mancanze o lacune per fornire piena soddisfazione;
- Sia valido, desiderabile, efficace, sostenibile nel tempo, utilizzabile, fattibile e sicuro sotto ogni aspetto;
- Sia il risultato del lavoro di alta professionalità;

- Incluva Visione, Strategia e Obiettivo di Prodotto convincenti, chiari e orientati ai risultati, che spesso includono intenzioni, motivazioni, oltre alla chiarezza sul cosa non va fatto;
- Si adatti e migliori per identificare, rappresentare o misurare il cambiamento emergente [21]; e
- Sia espandibile e di facile manutenzione.

In estrema sintesi, in Scrum il Prodotto è la manifestazione del *perché* facciamo *ciò* che facciamo.

Commitment: La Definition of Outcome Done La Definition of Outcome Done (“Definizione di Risultato Ottenuto”). Nuovo termine introdotto in questo pacchetto di Espansione, che d’ora in poi verrà lasciato in lingua Inglese, N.d.T.) è un impegno (commitment). Descrive le misurazioni di evidenza osservabili (quantitative o qualitative) necessarie per dimostrare i benefici conseguiti, spesso definite come validazione del Valore. Può riguardare il Prodotto nel suo complesso o limitarsi ad un obiettivo specifico. È consigliabile concordare e definire le unità di misura da impiegare per la validazione del Valore erogato o prodotto prima dell’inizio lavori, per prevenire qualsiasi distorsione o successivamente interpretazioni fuorvianti o errate.

I risultati (Outcomes) confermano i cambiamenti negli utenti; l’impatto conferma i risultati a livello organizzativo. Tutto questo, assieme alle relative interpretazioni, guida gli adattamenti futuri, idealmente confermando l’impatto previsto sugli Stakeholder (incluso, a titolo esemplificativo ma non esaustivo, l’impatto sul Business o sugli utenti), misurando se quanto realizzato soddisfa i risultati attesi e genera un Valore reale. Potrebbe riguardare un obiettivo specifico, come una o più funzionalità più ampie, ed essere convalidato tramite la telemetria del Prodotto (il Prodotto può misurare il proprio utilizzo). In alternativa, potrebbe riguardare il Prodotto nel suo complesso, dove l’attenzione spesso è rivolta all’impatto strategico e la convalida dell’efficacia dell’implementazione strategica [50-54]. O una combinazione di entrambi.

È cruciale preferire l’evidenza diretta rispetto all’evidenza circostanziale. Ad esempio:

- I risultati (Outcomes) fruiti dai Clienti potrebbero concentrarsi sulla fornitura del Valore misurabile prodotto, come maggiore soddisfazione, riduzione dei costi a lungo termine o il numero delle attività gestite.
- I risultati fruiti dagli utenti potrebbero riguardare cambiamenti specifici nel loro comportamento, che ne risolvono i problemi e ne migliorano l’esperienza (col Prodotto / Soluzione, N.d.T.), come ad esempio il completamento di attività in modo più efficiente o l’utilizzo di nuove funzionalità.
- I risultati fruiti dagli stakeholder di Prodotto/Soluzione potrebbero collegare questi cambiamenti comportamentali alle metriche di performance, come ad esempio, trend nel numero dei Clienti o comportamenti nell’uso del prodotto, metriche relative a decisori e/o utenti, frequenza e tempistica di rilascio del prodotto, tempi di apprendimento, tempi di cambiamento, ecc.
- L’impatto sul business, ad esempio, conformità, riduzione dei costi aziendali a lungo termine, risultati aziendali, trend nella quota di Mercato, soddisfazione

del Cliente riguardo alla gamma dei prodotti, tempi tecnico-operativi di rilascio, tempi di apprendimento, tempi di cambiamento, ecc.

- I risultati frutti dal Team Scrum, come miglioramento delle loro competenze e capacità tecniche (flusso psicologico [70], frequenza di rilascio, strumenti, competenze, Debito Tecnico (Technical Debt), compresi quelli per UX o CX, capacità di lavoro potenziale), clima/cultura relativi al miglioramento e l'innovazione.

Il Debito Tecnico (Technical Debt) relativo all'esperienza utente (UX) o all'esperienza cliente (CX) è il risultato cumulato di scelte errate di progettazione e/o implementazione, intenzionali o meno, che rendono un Prodotto o servizio progressivamente meno usabile, piacevole o efficace per gli utenti o i clienti. Riconoscere, monitorare e affrontare questo particolare tipo di “debito tecnico” è essenziale per realizzare un Prodotto o servizio che soddisfi davvero le esigenze e le aspettative degli utenti.

Le misurazioni effettuate progressivamente rendono trasparenti i trend di percezione di prodotto, di mercato e per tutti gli Stakeholder coinvolti (inclusi, a titolo esemplificativo ma non esaustivo, Clienti o utenti); questi possono essere esaminati in qualsiasi momento durante lo Sprint, incluso durante la Sprint Review.

Incremento

L'Incremento è un artefatto. Rappresenta l'integrazione del lavoro completato secondo gli standard della Definition of Output Done. L'Incremento è un risultato conforme, un output e un Prodotto potenzialmente rilasciabile.

È possibile creare più Incrementi all'interno dello stesso Sprint, grazie al completamento degli elementi del Product Backlog (PB Items). Ogni Incremento viene accuratamente verificato, reso utilizzabile e integrato con tutti gli Incrementi precedenti. L'Incremento aggregato risultante viene ispezionato il prima possibile durante lo Sprint, al più tardi durante la Sprint Review. L'Incremento deve essere utilizzabile e utile, permettendo il feedback sui risultati generati. Gli Incrementi sono elementi fondamentali per Scrum, in quanto consentono la convalida continua del Valore generato dal Team Scrum.

Un candidato di Incremento non è considerato tale finché non soddisfa gli standard di Qualità della Definition of Output Done. Un solo Incremento può essere rilasciato ogni volta, indipendentemente dalla sua frequenza. Un Incremento dovrebbe essere considerato una pietra miliare concreta verso l'Obiettivo del Prodotto. Gli Incrementi possono essere consegnati agli Stakeholder o rilasciati prima della Revisione dello Sprint. *La migliore validazione del Valore deriva infatti da un feedback sui risultati generati.*

Commitment: Definition of Output Done La Definition of Output Done (“Definizione di come il lavoro va completato”). Nuovo termine introdotto da questo pacchetto di Espansione, che d'ora in poi verrà lasciato in Inglese, N.d.T.) è un impegno (commitment). Descrive formalmente i parametri di qualità che esprimono la diligenza necessaria a realizzare l'Incremento, in modo che possa essere consegnato agli Stakeholder.

La Definition of Output Done tipicamente include (ma non è limitata a) gli standard tecnici e le qualità intrinseche ed estrinseche del Prodotto. Qualora non sia fornita dall'organizzazione, Il Team Scrum è tenuto a creare una propria Definition of Output Done. Se coesistono più Team Scrum sullo stesso Prodotto, questi condividono una Definition of Output Done comune, ma possono apportare miglioramenti alla propria.

Ogni Team Scrum è obbligato a conformarsi alla Definition of Output Done e a migliorarlo continuamente. L'Incremento è per sua natura cumulativo. La Definition of Output Done è a beneficio del Prodotto e dei suoi Stakeholder. La Definition of Output Done è lo standard di Qualità complessivo per l'intero Incremento, e non già uno standard specifico per ciascun elemento, come sarebbe ad esempio, per Criteri di Accettazione (Acceptance Criteria).

Solo una volta che l'Incremento viene rilasciato, è possibile attivare il feedback sui risultati, per la continua validazione della Definition of Outcome Done.

Product Backlog

Il Product Backlog è un artefatto di Scrum. È l'elenco emergente ed ordinato (in sequenza) degli Elementi del Product Backlog necessari per raggiungere l'Obiettivo di Prodotto. Il Product Backlog fornisce Trasparenza (chiarezza del lavoro) ed è l'unica fonte di lavoro per il Team Scrum per conseguire l'Obiettivo del Prodotto. Il Product Owner, tenendo sempre presente il Valore da generare, è responsabile dell'ordinamento degli Elementi del Product Backlog. Un Product Backlog di dimensioni ridotte spesso offre maggiore Trasparenza, nonché aiuta a ridurre il tempo di completamento.

Elemento del Product Backlog Con Elemento del Product Backlog si indica un elemento potenzialmente prezioso che compone il Product Backlog. Non è necessario che abbia un formato specifico e mira a gestire un problema o cogliere un'opportunità. Può eventualmente includere anche Criteri di Accettazione (Acceptance Criteria) che possono aiutare a determinare quando il lavoro è completato, mentre la Definition of Output Done è obbligatoria. Infatti, pur consegnando esattamente quanto richiesto, si potrebbe non fornire risultati sufficienti. Perciò, oltre a quanto deciso nella Definition of Outcome Done, includere anche Criteri di Risultato (Outcome Criteria) chiaramente definiti può aiutare a valutare se un Elemento del Product Backlog abbia fornito un valore sufficiente.

Un Elemento del Product Backlog è composto da una singola unità di lavoro, volta a scoprire o generare Valore. Un Elemento del Product Backlog può evolversi in qualsiasi momento, anche mentre i Developer ci lavorano. Durante le attività di Refinement il Product Backlog viene scomposto in Elementi del Product Backlog indipendenti (e più comprensibili, per lo più per il Team Scrum) che potrebbero potenzialmente fornire valore. Occasionalmente, un Elemento nel Product Backlog potrebbe non essere correlato direttamente all'Obiettivo del Prodotto (Product Goal); se ciò accade frequentemente, vale allora la pena esaminare se il livello di *Focus* potrebbe non essere nello stato ideale. Il Team Scrum e gli Stakeholder dovrebbero *Focalizzarsi* sugli esiti generati, piuttosto che sui Prodotti in sé, mantenere il giusto equilibrio tra i vari compromessi necessa-

ri, evitando che il Team Scrum diventi una ‘fabbrica di funzionalità (Feature)’ o ‘di scoperta (Discovery).’

Criteri di Accettazione Se presenti, i Criteri di Accettazione (Acceptance Criteria) descrivono quando il risultato di uno specifico Elemento del Product Backlog può considerarsi completo, oltre a quanto già definito nella Definition of Done. Negli elementi del Product Backlog con un livello di dettaglio sufficiente, i Criteri di Accettazione devono fornire chiarezza inequivocabile su *cosa* è richiesto. Essi includono aspetti specifici di un elemento del Product Backlog eventualmente non già coperti dalla Definition of Done e possono essere sia funzionali sia non funzionali. I Criteri di Accettazione possono essere aggiornati in qualsiasi momento, anche mentre i Product Developers stanno sviluppando.

Come parte di un insieme più ampio, i Criteri di Risultato (Outcome Criteria), qualora impiegati in quanto non obbligatori, solitamente descrivono l’intenzione dell’Elemento del Backlog del Prodotto; si tratta del *perché* che sta dietro il *cosa*. Il raggiungimento di questi criteri spesso integra la Definition of Outcome Done per il Prodotto. Possono includere criteri specifici per un Elemento del Backlog del Prodotto che non sono parte integrante della Definition of Outcome Done. In caso di dubbi, i criteri di risultato forniscono indicazioni supplementari o integrative; possono assumere la forma di una narrazione oppure essere rappresentati da misure, con preferenza per le seconde. I criteri di risultato possono evolvere in qualsiasi momento, anche mentre i Developers stanno lavorando.

Refinement Il Refinement è un’attività. Può essere formale (come un evento aggiuntivo) (in SGEP rappresenta un elemento chiave) o informale. Si tratta di un processo continuo ed emergente che favorisce la chiarezza e riduce i rischi; determina un livello sufficiente di comprensione e fiducia affinché gli elementi del Product Backlog già selezionati o imminenti siano “pronti” (ossia completabili in accordo con la Definition of Output Done entro un numero ridotto di giorni, o anche prima). Vengono anche prese in considerazione le varie tipologie di possibili dipendenze.

Il Refinement consiste nello scomporre gli elementi del Product Backlog in parti più piccole e comprensibili (soprattutto per il Team Scrum). Possono essere aggiunti ulteriori dettagli come descrizione, Criteri di Accettazione, ordine e dimensione. Gli attributi possono variare, ma dovrebbero avere significato per il Team Scrum. Il Refinement può includere attività di ricerca, tra cui – ma non solo – validazione di problemi o opportunità, esperienze di utenti o clienti, validazione della soluzione. Esclusivamente i Product Developers - e nessun altro - sono responsabili (*responsible*) del dimensionamento degli elementi del Product Backlog. Il Product Owner può indirizzare i Product Developers aiutandoli a comprendere meglio e selezionare i compromessi potenzialmente necessari.

Al Refinement partecipano spesso sia gli Stakeholder sia i membri del Team Scrum; non è raro che i Product Developers lavorino direttamente con gli Stakeholder. Il Refinement è spesso supportato o facilitato dal Product Owner. Il Product Owner può concentrarsi (*Focus*) maggiormente sulla Product Ownership se i Product Developers hanno

una solida comprensione del Prodotto. In generale, si tratta di un’attività orientata al futuro che offre chiarezza, direzione e un potenziale *Focus* per gli Sprint successivi.

Impegno (Commitment): Il Product Goal Il Product Goal è un impegno. È rappresentato attraverso il Product Backlog, che è di proprietà del Product Owner. Rappresenta l’obiettivo attuale, unico, più strategico e ambizioso (il *perché*). Fornisce direzione al prodotto e permette di mantenere il *Focus* per i Product Developers che lavorano sul Prodotto. Migliora la Trasparenza, offrendo una direzione chiara e di valore verso cui i Product Developers possono orientare il loro lavoro, utilizzando uno Sprint Goal più tattico (il *perché* dello Sprint).

Il Product Goal costituisce l’obiettivo a medio termine per il Team Scrum e per gli Stakeholder (e i Sostenitori). Il Team Scrum dovrebbe completare (o abbandonare) un Product Goal prima di affrontare il successivo.

Un Product Goal è solitamente un’affermazione ancora da verificare riguardo al valore. Può essere espresso in diversi modi, tra cui un insieme di ipotesi volte a colmare o ridurre i gap di soddisfazione (outcome) e l’impatto atteso sul business. Trova il giusto equilibrio concentrandosi su un sottoinsieme delle molteplici aspettative e vincoli degli Stakeholder (inclusi, ma non solo, clienti o utenti). Attraverso Ispezione e Adattamento, è fondamentale accogliere l’incertezza [41], i feedback sui risultati, gli effetti collaterali e altri apprendimenti.

Che dire della Product Vision? Molte organizzazioni lavorano con una Product Vision, che aiuta a visualizzare un futuro potenziale. Il Team Scrum può utilizzare una Vision come input per definire un Product Goal. Una Product Vision è un insieme significativo e di lungo termine di risultati desiderati e di valore. Il Product Goal a medio termine è spesso un passo intermedio verso una Product Vision di lungo periodo. (presente in SGEP)

Man mano che il Team Scrum e gli Stakeholder ispezionano e adattano il lavoro verso il Product Goal, è importante rimanere aperti (*open*) all’eventualità che anche la Product Vision o il Product Goal possano necessitare di adattamenti. È frequente che più Product Goal vengano completati uno dopo l’altro mentre si avanza verso una vision.

È importante notare che una Product Vision è spesso più un’ipotesi che una realtà concreta; nulla di tutto questo è necessariamente vero. È fondamentale formulare ipotesi e condurre esperimenti in una direzione precisa, ed è qui che Scrum può aggiungere il massimo valore.

Una Product Vision è spesso fonte di ispirazione, ma può anche risultare travolgenti. Il Product Goal attenua questa sensazione di sovraccarico, agendo da rappresentazione più concreta e tangibile della Product Vision o come elemento che ne facilita il raggiungimento.

Sprint Backlog

Lo Sprint Backlog è un artefatto. È composto dallo Sprint Goal (il *perché* dello Sprint), dall’insieme degli elementi del Product Backlog selezionati per lo Sprint (il *cosa*, noto anche come previsione) e spesso da un piano operativo per realizzare l’Incremento (il *come*). Fornisce Trasparenza (chiarezza sul lavoro) durante tutto lo Sprint.

Lo Sprint Backlog è un piano dei e per i Product Developers. Rappresenta il loro punto di vista sul lavoro compreso e necessario per raggiungere lo Sprint Goal (il perché dello Sprint). Immaginiamo uno scenario non ottimale in cui la maggior parte degli elementi presenti nello Sprint Backlog risultino continuamente non collegati al Product Goal: in tal caso i Valori Scrum di *Focus* e *Commitment* non vengono rispettati.

Nel contesto dello Sprint Goal, i Product Developers aggiornano il loro piano, inclusa la previsione, per tutta la durata dello Sprint man mano che apprendono nuove informazioni. Lo Sprint Backlog dovrebbe contenere abbastanza lavoro per poter iniziare, ad esempio con uno o due elementi del Product Backlog utili al raggiungimento dello Sprint Goal. (Idea della SG2017 ripresa) I Product Developers ispezionano i propri progressi verso lo Sprint Goal nel Daily Scrum, o anche più frequentemente. In questo modo imparano ad adattarsi e a rispondere all’incertezza [41].

Commitment: Lo Sprint Goal Per garantire coerenza, lo Sprint Goal è un impegno creato e di responsabilità del Team Scrum (Nella SG2020, i Developer erano responsabili del commitment). È l’obiettivo che guida e unifica lo Sprint (il *perché*) per i Product Developers ed è definito durante lo Sprint Planning. Il raggiungimento dello Sprint Goal è un impegno dei Product Developers. Lo Sprint Backlog (inclusi il *perché*, il *cosa* e, spesso, il *come*) fornisce *Focus* e flessibilità nell’adattarsi al lavoro in corso di sviluppo, migliorando così la trasparenza.

Lo Sprint Goal incoraggia il Team Scrum a lavorare insieme piuttosto che su iniziative separate. Se il lavoro si rivela diverso da quanto previsto dai Product Developers, questi collaborano con il Product Owner per negoziare possibili adattamenti all’interno dello Sprint, senza compromettere lo Sprint Goal. Nessuno indica ai Product Developers come dimensionare o svolgere il proprio lavoro.

Se ci sono più obiettivi, purché siano coerenti con il Product Goal, può andare bene. Tuttavia, è necessario prestare attenzione. Occorre considerare il compromesso tra context switch e focus. Il team Scrum dovrebbe sforzarsi di raggiungere un equilibrio nel tempo, bilanciando le capacità e il clima attuali del team, il valore attuale per gli stakeholder, il potenziale valore per gli stakeholder e il time to value. (SGEP prende in considerazione DevSecOps)

Gli Eventi Scrum considerati nel Pacchetto di Espansione

Scrum combina quattro eventi di durata limitata (detti “timebox”) necessari per Ispezione e Adattamento, che si svolgono all’interno di un quinto evento di durata fissa, lo Sprint che li raggruppa. Questi eventi supportano i pilastri Scrum di Trasparenza,

Ispezione e Adattamento. I rilasci del prodotto servono a generare Valore, idealmente, in modo continuo. Rilasci poco frequenti comportano ritardi nei feedback sui risultati.

Un timebox è un intervallo di tempo di durata massima fissa, stabilito dall'inizio alla fine di un determinato evento: non è necessario utilizzare l'intero intervallo di tempo. Lo scopo dei timebox in Scrum è quello di favorire la selezione del lavoro effettivamente essenziale, creando *Focalizzazione* (Focus) per raggiungere rapidamente i risultati desiderati.

Gli eventi favoriscono una cadenza regolare e riducono al minimo la necessità di altre riunioni, che infatti non sono previste in Scrum. Idealmente, ogni evento si tiene nello stesso orario e nello stesso luogo per ridurre la complessità [12-17] e favorisce la formazione di buone abitudini. Per aumentare l'efficacia della facilitazione, è consigliabile avere un facilitatore qualificato ed esperto. Infatti, eventi inefficaci rischiano di far perdere al Team l'enfasi sullo Sprint Goal, sul Product Goal, nonché sui 3 pilastri - Trasparenza, Ispezione e Adattamento - nonché sui 5 Valori di Scrum - Impegno, Focus, Apertura, Rispetto e Coraggio.

Ogni evento ha un proprio scopo preciso e dovrebbe includere attività approfondite e significative. Insieme, gli eventi Scrum forniscono la necessaria "impalcatura" per generare la Trasparenza necessaria per fermarsi a riflettere, analizzare, per poi apportare adattamenti. Gli eventi Scrum favoriscono pensiero e lavoro strutturati, incrementano l'efficacia e aiutano a determinare un carico di lavoro equilibrato. (SGEP si ispira a Cynefin e all'Evidence-Based Management)

Una comunicazione continua ed efficace è fondamentale per garantire che il Team Scrum e coloro che lo supportano si concentrino sul fare la cosa giusta. Con la sola eccezione dello Sprint (una volta definita la sua durata, N.d.T.), gli altri quattro eventi possono avere una durata inferiore, a patto che non se ne perda la coerenza ed efficacia.

Lo Sprint

Lo Sprint è un evento in cui le idee si trasformano in Valore. Lo Sprint è l'evento che racchiude tutti gli altri eventi di Scrum. È un'iterazione di durata determinata, durante la quale viene effettivamente svolto il lavoro. Fornisce *Focalizzazione* (Focus) e stabilità. Uno Sprint non può durare più di quattro settimane. Il nuovo Sprint inizia immediatamente dopo il termine di quello precedente. Tutto il lavoro necessario per raggiungere il Product Goal si svolge esclusivamente all'interno degli Sprint.

Gli Sprint sono il "battito" del cuore di Scrum, durante i quali il Team Scrum trasforma le idee in Incrementi utilizzabili, utili e potenzialmente di valore tangibile. L'Incremento viene rilasciato il prima possibile, in considerazione della necessità di un tempestivo feedback sui risultati generati. Il mancato rilascio ad alcuni sottogruppi di Stakeholder (inclusi clienti, decisori e utenti - a titolo esemplificativo ma non esaustivo) può determinare la mancanza di feedback tempestivo sui risultati. In uno Sprint è certamente possibile creare più Incrementi; il Team Scrum infatti - quando possibile - dovrebbe impegnarsi a validare il Valore generato attraverso rilasci anticipati, frequenti e tempestivi.

Durante lo Sprint:

- Non vengono apportate modifiche che potrebbero mettere a repentaglio lo Sprint Goal;
- La qualità dell'incremento non deve essere diminuita o compromessa;
- Si perfeziona il Product Backlog a seconda delle necessità;
- Man mano che più informazioni vengono raccolte, il lavoro in corso può essere meglio chiarito ed eventualmente rinegoziato con il Product Owner, senza però compromettere lo Sprint Goal.

Gli Sprint permettono il raggiungimento dei risultati desiderati garantendo Ispezione e Adattamento dei progressi verso lo Sprint Goal almeno ogni quattro settimane. Se si sceglie una durata dello Sprint troppo lunga, lo Sprint Goal potrebbe diventare incoerente, aumentando complessità [12-17] e rischio. Sprint di durata più breve spesso generano un numero maggiore di cicli di apprendimento e potrebbero anche contribuire a limitare il rischio.

Sprint più brevi richiedono solitamente capacità superiori e consolidate (ad esempio, refinement, slicing verticale, migliore dominio degli aspetti tecnici, dominio aziendale). Dato che adattarsi al contesto è importante, il Team Scrum si impegna a trovare pragmaticamente il giusto equilibrio.

Esistono diverse pratiche complementari per valutare o prevedere (forecast) i progressi, come grafici burn-down o burn-up, analisi del Flow, previsioni probabilistiche basate su simulazioni Monte Carlo, stime di ampio impatto (large effort estimation), fuzzy sets [55], ecc. Sebbene utili, queste pratiche o tecniche non possono mai sostituire l'importanza dell'Empirismo [19]. Infatti negli ambiti definiti "complessi" [12-17], ciò che è già accaduto può essere utilizzato per processi decisionali lungimiranti, ma deve esserci piena consapevolezza che quanto accadrà in futuro è comunque sempre sconosciuto.

Sebbene si potrebbe pensare a uno Sprint come a un "mini progetto" con un risultato atteso, durata e costi definiti, in Scrum le varie attività di lavoro si svolgono in parallelo e non in modo sequenziale e lineare.

Qualora lo Sprint Goal diventasse obsoleto, lo Sprint può essere annullato e solamente il Product Owner ha l'autorità per farlo. Sprint più brevi aiutano a ridurre la probabilità di annullamento.

Sprint Planning

Lo Sprint Planning è un evento. Essendo il primo evento dello Sprint è il momento in cui il Team Scrum pone l'attenzione su *Focalizzazione* (Focus) e *Impegno* (Commitment).

Durante lo Sprint Planning, il Product Goal strategico (il *perché* del Product Backlog) viene considerato e fornisce la direzione da seguire. In questo modo, i Developers del Prodotto creano lo Sprint Backlog, che consiste nell'obiettivo di breve termine, più tattico (il *perché* dello Sprint), nel lavoro inizialmente identificato e nello Sprint Plan, o piano di consegna.

Lo Sprint Planning affronta i seguenti argomenti:

Il Perché dello Sprint Il Product Owner propone idee supportate dai dati (ad es. nel Refinement, attraverso feedback sui risultati o altri apprendimenti) su come il Prodotto potrebbe aumentare il Valore erogato e la sua utilità nello Sprint appena iniziato. Il Product Owner e i Product Developer collaborano quindi per definire uno Sprint Goal che comunichi perché lo Sprint è prezioso per gli Stakeholder in relazione al Product Goal (SGEP enfatizza la pianificazione direzionale e just-in-time. Lo Sprint Goal deve essere finalizzato entro la fine dello Sprint Planning).

Il Cosa verso il Perché In stretta collaborazione con il Product Owner, i Developers selezionano gli elementi dal Product Backlog da includere nello Sprint in corso. Il Team Scrum può perfezionare questi elementi, aumentandone la comprensione e la sicurezza. Gli elementi selezionati devono essere realizzabili secondo gli standard concordati nella Definition of Output Done, insieme ad altri elementi.

Stabilire quanto può essere completato in uno Sprint può essere decisamente difficile. Tuttavia, più i Developers conoscono le performance passate, adottano il vertical slicing, valutano la potenziale capacità futura e affinano la Definition of Output Done, maggiore sarà la loro confidenza nella previsione dei risultati dello Sprint.

I Team Scrum di successo evitano di sovraccaricarsi. Anzi, pianificano di terminare il lavoro in anticipo, considerando a volte un margine di manovra (buffer) per eventi imprevisti [56]. Questo aiuta il Team Scrum a rimanere concentrato, migliorare la qualità e soddisfare gli Stakeholder offrendo Valore in tempi più rapidi. Sovraccarichi cronici o cambiamenti improvvisi possono causare un eccessivo stress negativo, che Jeff Sutherland definisce “sorpresa Bayesiana”. Possono compromettere il flusso psicologico [46] e le prestazioni del Team Scrum. Una comunicazione chiara, una gestione professionale del lavoro emergente [21] e piccoli cambiamenti regolari aiutano a prevenire questo problema, quindi i Team Scrum dovrebbero puntare a rilasciare il più possibile anticipatamente. (SGEP favorisce tempi di slack)

Il Come per il Cosa Le modalità di esecuzione del lavoro sono a esclusiva discrezione dei Developers. Nessun altro spiega ai Developers come svolgere il proprio lavoro. I Developers selezionano autonomamente il proprio lavoro; nessun altro assegna o inoltra elementi del Product Backlog ai Developers, nemmeno il Product Owner.

Lo Sprint Planning è limitato a un massimo di otto ore per uno Sprint della durata di quattro settimane. L'evento è solitamente più breve per Sprint più brevi. Il contesto è fondamentale, ma come regola generale è consigliabile pianificare attività sufficienti per iniziare il lavoro, ad esempio iniziando a pianificare solo alcuni elementi del Product Backlog, inizialmente utili per raggiungere lo Sprint Goal.

Daily Scrum

Il Daily Scrum è un evento. Durante il Daily Scrum, i Developers collaborano per raggiungere lo Sprint Goal e aggiornano il loro piano d'azione - rappresentato dallo

Sprint Backlog - fino al Daily Scrum successivo. Nel caso in cui lo Sprint Goal sia già stato raggiunto, i Developers collaborano per raggiungere altri progressi significativi aggiuntivi verso il Product Goal. (SGEP enfatizza l'importanza della direzione)

Il Daily Scrum fornisce *Focalizzazione* (Focus), coesione e urgenza e promuove l'auto-organizzazione [11]. Di solito, è necessaria la partecipazione dei soli Developers. Per mantenere le cose semplici, per il Daily Scrum si preferisce mantenere costanti cadenza, luogo e orario delle riunioni.

I Developers possono scegliere la struttura e le tecniche che preferiscono. I Daily Scrum migliorano la comunicazione per raggiungere lo Sprint Goal, identificano e affrontano rischi e ostacoli tempestivamente, promuovono un processo decisionale rapido, eliminando di conseguenza la necessità di altre riunioni.

Il Daily Scrum non è l'unico momento in cui i Developers modificano il loro piano per lo Sprint nel contesto dello Sprint Goal o del Product Goal. I Developers si incontrano spesso durante il giorno per discussioni più approfondite.

Per favorire il Flusso del Valore [48-49] e ottenere risultati potenziali in tempi più rapidi, i Developers dovrebbero concentrarsi su uno o più elementi alla volta e soddisfare la Definition of Output Done, prima di iniziare a lavorare su altri elementi. I Developers possono raggiungere questo obiettivo aumentando la Focalizzazione, riducendo il numero di processi in corso e preferendo terminare il lavoro in corso anziché aggiungerne di nuovi. I Developers tengono sotto controllo il lavoro inattivo e non le persone inattive. (SGEP enfatizza il flusso)

La durata del Daily Scrum è limitata a quindici minuti al giorno.

Sprint Review

La Sprint Review (Revisione dello Sprint) è un evento. È una sessione di lavoro interattiva e collaborativa. Spesso, il Team Scrum condivide il Product Goal corrente e presenta la Definition of Output Done e la Definition of Outcome Done agli Stakeholder. Il Team Scrum condivide i risultati del proprio lavoro, i compromessi raggiunti e i progressi ottenuti verso il Product Goal (il *perché* del lavoro). Qualora siano disponibili, vengono anche condivise e considerate le misurazioni e/o le metriche correnti, adottate per valutare i progressi verso la Definizione di Outcome e considerare se eventualmente vadano aggiornate.

La Sprint Review esamina molti aspetti correlati al Prodotto, come Product Goal, il Product Backlog, lo Sprint Goal, quanto appreso durante il lavoro, l'incremento, le aspettative e i limiti degli Stakeholder come parti interessate, il feedback sui risultati, gli effetti collaterali, i progressi del Prodotto, la situazione del Mercato di riferimento, nonché aspetti lungimiranti, ad esempio quali nuove idee e opportunità che sono emerse durante lo Sprint e i potenziali passi successivi.

Informati da quanto appreso, i partecipanti:

- percepiscono, ascoltano, imparano e collaborano su cosa potenzialmente fare in seguito;

- adattano il Product Backlog (il cosa) e possibilmente anche il Product Goal idealmente supportato da evidenze o osservazioni e guidato dal Product Goal o dalla eventuale Product Vision;
- adattano la Definition di Outcome Done del prodotto per gli Sprint futuri.

Durante la Sprint Review è opportuno considerare attentamente ciò che viene ritenuto importante da tutti i soggetti coinvolti o le parti interessate, comprese quelle inanimate o non umane come, ad esempio, la Legislazione vigente.

Eventuali elementi del Product Backlog non completati durante lo Sprint tornano a far parte del Product Backlog per una eventuale valutazione futura e, essendo incompleti, non vengono presentati durante la Sprint Review; talvolta, vengono spostati direttamente nello Sprint successivo (come parte dello Sprint Goal, N.d.T.).

La Sprint Review è il penultimo evento dello Sprint e ha una durata limitata a quattro ore per uno Sprint di quattro settimane; per Sprint più brevi, l'evento è solitamente più breve.

Sprint Retrospective

La Sprint Retrospective (o Retrospettiva) è un evento. In questo evento, il Team Scrum concorda su come migliorarsi. Vengono inoltre esplorate le ipotesi errate, ovvero quelle che possono aver portato il Team Scrum nella direzione sbagliata. Aspetti positivi come tecnologie, processi, elementi ricorrenti, ecc. possono essere evidenziati o rafforzati. Gli aspetti sottoposti ad ispezione e analisi variano spesso a seconda del settore di lavoro. Il processo retrospettivo diventa più efficace in presenza di un ambiente psicologicamente sicuro.

La Sprint Retrospective si concentra sui possibili cambiamenti, che possono generare miglioramenti riguardo a:

- Incremento (come risultato dello Sprint)
- Outcomes (Risultati tangibili) e impatto sul business
- Professionalità del team, ad esempio competenze, pratiche tecniche, strumenti, capacità di innovare;
- Flusso di Valore convalidato [48-49], ad esempio, end-to-end Flow Metrics (misurazione efficienza del flusso completo), time-to-market;
- Efficacia (il *come*), ad esempio tecnologia, processi, vincoli e dipendenze;
- Interazioni e dinamiche del team Scrum, ad esempio collaborazione, accordi di lavoro;
- Information Radiators (supporti visivo per condividere informazioni, N.d.T.) come ad esempio Board, Product Wall o metriche;
- La definizione dell'output realizzato per gli Sprint futuri;
- Ulteriori adattamenti alla Definition of Output Done per gli Sprint futuri;
- Come calcolare automaticamente le metriche relative alla Definition of Output Done;
- e altro ancora.

I miglioramenti più significativi dovrebbero essere affrontati il prima possibile. Il Team Scrum non dovrebbe limitarsi a discutere solamente degli eventuali miglioramenti; Scrum si basa su un miglioramento continuo e significativo. Anche se alcune azioni di miglioramento potrebbero basarsi sul supporto degli Stakeholder di tipo Supporter, ciò non significa che il Team Scrum non debba comunque impegnarsi autonomamente per ottenere miglioramenti netti (come ad esempio continui progressi marginali).

La Sprint Retrospective conclude lo Sprint. Ha una durata limitata a tre ore per uno Sprint di quattro settimane; per Sprint più brevi, l'evento è solitamente più breve.

Note conclusive

Scrum è come uno specchio: se l'immagine riflessa non è quella attesa, dovrresti forse coprire lo specchio?

Assicurati di completare almeno un Incremento per ogni Sprint come abitudine prima di adattare Scrum alle specifiche esigenze. Ogni elemento di Scrum ha uno scopo; comprendere il perché di ogni suo elemento è essenziale. Valuta il contesto di adozione: nel breve termine punta alla consegna di Valore, mentre per il lungo periodo focalizzati sul successo del cambiamento emergente in una determinata direzione nonché a rendere sostenibile l'erogazione di Valore: outcome per gli Stakeholders e impatto per l'organizzazione. Il successo nell'adozione di Scrum dipende dal raggiungimento del giusto equilibrio tra breve e lungo termine.

Fai attenzione a non copiare approcci da altre organizzazioni senza promuoverne anche la cultura. Punta al vero cambiamento, ovvero quello emergente dal percorso nella direzione di marcia concordata. Il cambiamento richiede di operare (ma non limitatamente a) su Leadership, flussi di lavoro, processi e sistemi, cui Risorse Umane, Finanza, Acquisti e altro ancora. Scrum fa parte di un percorso - idealmente senza una fine predefinita - di miglioramento continuo ed evoluzione con una data direzione di marcia, piuttosto che avere una vera e propria destinazione.

Crediti e riconoscimenti

Scrum trae ispirazione da Lean [20], dal Sistema di Produzione di Toyota (Toyota Production System) [57-58], dall'articolo apparso su Harvard Business Review intitolato 'The New New Product Development Game' di Hirotaka Takeuchi e Ikujiro Nonaka [18], nonché dall'Empirismo impiegato in Dupont [59].

Scrum venne sviluppato nei primi anni '90 da Ken Schwaber e Jeff Sutherland, che lo presentarono congiuntamente per la prima volta durante la conferenza OOPSLA Conference nel 1995 [60]. La prima versione della Guida a Scrum Scrum Guide [7] apparve nel 2009. Scrum è tuttora in continua evoluzione.

Vogliamo ringraziare i revisori che ci hanno donato i loro feedback alle prime versioni includendo, non limitati a, Daryn Basson, Alex Benes, Kurt Bittner, Deb Bhattacharya, Magdalena Firlit, Nichervan Fazel, Peter Fischbach, Michael Forni, Tom Gilb,

Martin Hinshelwood, Jesse Houwing, Michael Huynh, Matthew Ijogi, Marc Kaufmann, Christian Neverdal, Stas Pavlov, Ian Sharp, Alisa Stolze, Mark Summers e Nader Talai.

Attribuzione per la raccolta del Pacchetto di Espansione della Guida Scrum

Questa raccolta è stata scritta e compilata da *Ralph Jocham, John Coleman, e Jeff Sutherland*. La traduzione in lingua Italiana è stata curata da Michael F Forni, Felice Grisi e Andrea Feraco. Ogni sezione è attribuita individualmente sopra e mantiene la sua licenza originale. La raccolta nel suo complesso ha scopo informativo; si prega di rispettare i termini di licenza di ciascuna sezione.

Riferimenti

- [1] Langton, C.G. (ed.) (1989) Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, Los Alamos, New Mexico, September 1987. Santa Fe Institute Studies in the Sciences of Complexity, vol. VI. Redwood City, CA: Addison-Wesley.
- [2] Langton, C.G. (1989) ‘Life at the edge of chaos’, in Langton, C.G. (ed.) Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems. Santa Fe Institute Studies in the Sciences of Complexity, vol. VI. Redwood City, CA: Addison-Wesley, pp. 41–91.
- [3] Holland, J.H. (1992) Complex Adaptive Systems. *Daedalus*, 121(1), pp. 17–30. Available at: <https://www.jstor.org/stable/20025416> (Accessed: 17 May 2025).
- [4] Axelrod, R. and Cohen, M.D. (2000) Harnessing Complexity: Organizational Implications of a Scientific Frontier. New York: Free Press.
- [5] Juarrero, A. (1999) Dynamics in Action: Intentional Behavior as a Complex System. Cambridge, MA: MIT Press.
- [6] Snowden, D.J. and Boone, M.E. (2007) ‘A leader’s framework for decision making’, Harvard Business Review, 85(11), pp. 68–76. Available at: <https://hbr.org/2007/11/a-leaders-framework-for-decision-making> (Accessed: 17 May 2025)
- [7] Schwaber, K. & Sutherland, J. (2020) The Scrum Guide: The definitive guide to Scrum: The rules of the game. Available at: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf> (Accessed: 17 May 2025)
- [8] Schwaber, K. (2023) ‘Scrum Guide’, Ken Schwaber’s Blog, 25 September. Available at: <https://kenschwaber.wordpress.com/2023/09/25/scrum-guide/> (Accessed: 20 May 2025).
- [9] Mayer, T. (2025) A Simple Guide to Scrum. [Online]. Available at: <https://scrum.academy/guide/> (Accessed: 17 May 2025)
- [10] Scrum.org (2025) Scrum Hexis. Available at: https://thecynefin.co/product/hexi-scrumorg/?srsltid=AfmBOorcohLYeVy0qBsQFI6mK_bZtJA_uGC6hPL2BdptiTwNmMwpKTQv (Accessed: 17 May 2025).
- [11] LeSS.works (n.d.) Self-managing teams. Available at: <https://less.works/less/management/self-managing-teams> (Accessed: 17 May 2025).
- [12] Cynefin.io, V. (2022) Cynefin wiki, Cynefin.io. Cynefin.io. At: <https://cynefin.io/> (Accessed: April 4, 2023).

- [13] Rancati, A. and Snowden, D. (2021) Managing complexity (and chaos) in a crisis - a field guide for decision makers inspired by the Cynefin framework. Luxembourg, Belgium: Publications Office of the European Union.
- [14] Snowden, D. et al. (2022) Cynefin® weaving sense-making into the fabric of our world. 2nd edn. Edited by R. Greenberg and B. Bertsch. Singapore, Singapore: Cognitive Edge - The Cynefin Co.
- [15] Snowden, D. (2023) Cynefin St David's 2023 1 of 2, Cynefin Co. <https://thecynefin.co/cynefin-st-davids-2023-1-of-2/> (Accessed: April 20, 2023).
- [16] Snowden, D. (2023) Managing for emergence through abduction, The Cynefin Co. At: <https://thecynefin.co/managing-for-emergence/> (Accessed: June 24, 2023).
- [17] Snowden, D. and Smith, N. (2023) Leadership discussion: Dave and Natalie - the Cynefin co, YouTube. At: <https://youtu.be/WcPZ8ybDF0w> (Accessed: April 7, 2023).
- [18] Takeuchi, H. and Nonaka, I. (2014) The new new product development game, Harvard Business Review. At: <https://hbr.org/1986/01/the-new-new-product-development-game> (Accessed: 21 January 2024).
- [19] Campbell, J. (2025) 'Empiricism', EBSCO Research Starters. Available at: <https://www.ebsco.com/research-starters/religion-and-philosophy/empiricism> (Accessed: 17 May 2025)
- [20] Womack, J.P. and Jones, D.T. (1996) Lean Thinking: Banish Waste and Create Wealth in Your Corporation. New York: Simon & Schuster.
- [21] Templeton Foundation (2023) 'What Is Emergence?' John Templeton Foundation. Available at: <https://www.templeton.org/news/what-is-emergence> (Accessed: 17 May 2025).
- [22] Wolfram, S. (2002) A new kind of science. Champaign, IL: Wolfram Media.
- [23] Sutherland, J., Coplien, J.O., Heasman, L., den Hollander, M., Ramos, C. and The Scrum Patterns Group (2019) A Scrum Book: The Spirit of the Game. Raleigh, NC: Pragmatic Press.
- [24] Alexander, C. (1979) The timeless way of building. New York: Oxford University Press.
- [25] Lewis, R. (2023) An operating model for business agility: Agile for managers of the digital age. Independently published.
- [26] less.works (n.d.) Technical Excellence. Available at: <https://less.works/less/technical-excellence> (Accessed: 7 June 2025)
- [27] Thurlow, N., Turner, J.R. and Podder, A. (2020) The Flow System: The Evolution of Agile and Lean Thinking in an Age of Complexity. Flow Consortium. Available at: https://flowguides.org/Flow_Guide.pdf (Accessed: 17 May 2025).
- [28] Felderer, M. and Travassos, G.H. (2020) 'The Evolution of Empirical Methods in Software Engineering'. Available at: <https://arxiv.org/pdf/1912.11512.pdf> (Accessed: 17 May 2025).
- [29] Creative Wisdom (n.d.) 'Abduction, Deduction and Induction'. Available at: <https://www.creative-wisdom.com/teaching/WBI/abduction5.pdf> (Accessed: 17 May 2025).
- [30] Blackburn, S. (2003) Ethics: A Very Short Introduction. Oxford: Oxford University Press.
- [31] Cagan, M., 2018. Inspired: How to Create Tech Products Customers Love. 2nd

- ed. Hoboken, NJ: Wiley.
- [32] Cagan, M. & Jones, C., 2020. Empowered: Ordinary People, Extraordinary Products. Hoboken, NJ: Wiley.
- [33] Cagan, M., 2024. Transformed: Moving to the Product Operating Model. Hoboken, NJ: Wiley.
- [34] Moore, G.A., 1991. Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. New York: Harper Business.
- [35] Davies, Dan. (2025) The Unaccountability Machine: Why Big Systems Make Terrible Decisions—and How the World Lost Its Mind. London: Profile Books Ltd. (Paperback edition).
- [36] Ackoff, R.L. (1999) Ackoff's Best: His Classic Writings on Management. New York: John Wiley & Sons.
- [37] LeSS (n.d.) ‘Why LeSS? Achieving adaptiveness’. Available at: <https://less.works/less/framework/why-less> (Accessed: 17 May 2025).
- [38] Gothelf, J. & Seiden, J. (2021) Lean UX: Designing great products with agile teams. 3rd edn. Sebastopol, CA: O'Reilly Media
- [39] Torres, T. (2021) Continuous discovery habits: Discover products that create customer value and business value. North Charleston, SC: Product Talk
- [40] ‘Genchi Genbutsu’ (2024) Wikipedia. Available at: https://en.wikipedia.org/wiki/Genchi_Genbutsu (Accessed: 18 May 2025).
- [41] van der Bles, A.M., van der Linden, S., Freeman, A.L.J. and Spiegelhalter, D.J. (2019) ‘Communicating uncertainty about facts, numbers and science’, Royal Society Open Science, 6(5), 181870. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6549952/> (Accessed: 17 May 2025).
- [42] NetSuite (2023) ‘What Is Business to Business to Consumer (B2B2C)?’ Available at: <https://www.netsuite.com/portal/resource/articles/ecommerce/b2b2c.shtml> (Accessed: 17 May 2025).
- [43] Dictionary Marketing (2024) ‘B2B2B’. Available at: <https://dictionarymarketing.com/definition/b2b2b/> (Accessed: 17 May 2025).
- [44] Patton, J. and Economy, P. (2014) User Story Mapping: Discover the Whole Story, Build the Right Product. Sebastopol, CA: O'Reilly Media.
- [45] Kotter, J.P., 1996. Leading Change. Boston: Harvard Business School Press.
- [46] Csíkszentmihályi, M. (1990) Flow: The Psychology of Optimal Experience. New York: Harper & Row
- [47] Stripe (2025) ‘Sir Jony Ive and Patrick Collison Fireside Chat | Stripe Sessions 2025’, YouTube video, 8 May. Available at: https://youtu.be/wLb9g_8r-mE?si=1rEJxU0sxvblQ3&t=1390 (Accessed: 8 June 2025)
- [48] Kanban Guides (2025) Available at: <https://kanbanguides.org> (Accessed: 17 May 2025)
- [49] Scrum.org et al. (2021) The Kanban Guide for Scrum Teams. Available at: <https://www.scrum.org/resources/kanban-guide-scrum-teams> (Accessed: 17 May 2025)
- [50] Scotland, K. (2023) Why strategy deployment? Here are three great reasons, AvailAgility. At: <https://availagility.co.uk/2023/02/16/why-strategy-deployment-here-are-three-great-reasons/> (Accessed: April 3, 2023).
- [51] Scotland, K. (2019) Deploying strategies as choices, AvailAgility. At: <https://availagility.co.uk/2019/02/08/deploying-strategies-as-choices/> (Accessed: April 3,

2023).

- [52] Scotland, K. (2017) Strategy deployment and playing to win, AvailAgility. At: <https://availagility.co.uk/2017/07/14/strategy-deployment-and-playing-to-win/> (Accessed: April 3, 2023).
- [53] Scotland, K. (2017) A strategy deployment cadence, AvailAgility. At: <https://availagility.co.uk/2017/09/06/a-strategy-deployment-cadence/> (Accessed: April 3, 2023).
- [54] Scotland, K. (2022) The ultimate X-matrix for your agile transformation is here, AvailAgility. At: <https://availagility.co.uk/2022/11/03/the-ultimate-x-matrix-for-your-agile-transformation-is-here/> (Accessed: April 5, 2023).
- [55] Fuzzy Business: How to be roughly right rather than precisely wrong (unpublished).
- [56] ScrumPlop, n.d. Illigitimus Non Interruptis. The Scrum Book: The Spirit of the Game. Available at: <https://sites.google.com/a/scrumplop.org/published-patterns/product-organization-pattern-language/illegitimus-non-interruptus> [Accessed: 18 May 2025].
- [57] Ohno, T. (1988) Toyota Production System: Beyond Large-Scale Production. Portland, OR: Productivity Press.
- [58] Toyota Motor Corporation (2024) Toyota Production System. Available at: <https://global.toyota/en/company/vision-and-philosophy/production-system/index.html> (Accessed: 17 May 2025).
- [59] Hounshell, D.A. & Smith, J.K. (1988) Science and Corporate Strategy: DuPont R&D, 1902–1980. Cambridge: Cambridge University Press.
- [60] Schwaber, K. and Sutherland, J. (1995) ‘SCRUM Development Process’, OOPSLA Business Object Design and Implementation Workshop. Austin, Texas, October 1995. Available at: <http://jeffsutherland.org/oopsala/schwapub.pdf> (Accessed: 17 May 2025).
- [61] Members of The Scrum Patterns Group: Vervloed, E., Harrison, N., Harada, K., Yoder, J., Kim, J., O’Callaghan, A., Beedle, M., Bjørnvig, G., Friis, D., Reijonen, V., Benefield, G., Østergaard, J., Eloranta, V.-P., Leonard, E. & Aguiar, A.
- [62] Boyd, J.R. (1995–1996) The Essence of Winning and Losing. Unpublished briefing slides. Note: Boyd’s OODA was primarily disseminated through military briefings and unpublished manuscripts. His final conceptualization appears in The Essence of Winning and Losing, which emphasizes nonlinear decision-making and adaptation in complex environments.
- [63] Turner, J.R., Thurlow, N. and Rivera, B. (2019) The Flow System Guide. Available at: https://flowguides.org/Flow_Guide.pdf (Accessed: 24 May 2025). Summary: This guide integrates Boyd’s OODA with complexity theory and agile practices, framing it as a dynamic, non-linear decision-making process for organizational flow.