



# Scrum

Un insieme di best practice per affrontare lo sviluppo di un prodotto software (come per esempio un videogioco o una storia animata) in maniera agile e sempre in linea con le esigenze di un contesto in continuo cambiamento.

---

“Scrum è un **framework** che consente alle **persone** di risolvere problemi di tipo **adattivo** complessi e al tempo stesso di rilasciare prodotti di **altissimo valore** in maniera **efficace** e **creativa**” (*Scrum guide - Ken Schwaber e Jeff Sutherland*). Nella definizione degli autori della guida ufficiale a Scrum, è racchiusa l'essenza di questo strumento che centinaia di migliaia di team di progetto sparsi per il mondo impiegano ogni giorno per realizzare prodotti di valore legati all'ambito della programmazione del software.

Scrum non è un metodo bensì un **framework**, all'interno del quale è possibile utilizzare differenti tecniche e processi a seconda degli obiettivi che si desiderano raggiungere e delle risorse che si hanno a disposizione. Questo significa che Scrum non fornisce una struttura rigida di attività che devono essere eseguite pedissequamente, ma fornisce una serie di strumenti (*best practice*) tra cui scegliere quelli che meglio possono essere asserviti al raggiungimento dei propri obiettivi.

Scrum si realizza attraverso le **persone**: “*scrum*” è una parola presa a prestito dal mondo del rugby e sta ad indicare la “mischia” ovvero quella fase dell'incontro in cui l'arbitro ordina la ripresa del gioco tra due gruppi ordinati di giocatori appartenenti a squadre contrapposte. Così come durante questa fase i giocatori della medesima squadra si aiutano reciprocamente con l'obiettivo comune di entrare in possesso della palla e andare a meta, allo stesso modo i membri di un team di sviluppo comunicano e si aiutano reciprocamente per raggiungere gli obiettivi che si sono posti.

Scrum consente di risolvere problemi complessi in maniera **adattativa**: si basa sulla teoria del controllo empirico dei processi che afferma che la conoscenza deriva dall'esperienza e che le decisioni si possono basare soltanto su ciò che si conosce.

Questo tipo di approccio è andato diffondendosi negli ultimi anni come alternativa efficace ai modelli tradizionali di gestione dei progetti software risalenti agli anni '50, che prevedevano una pianificazione rigida delle attività e un approccio sequenziale alla progettazione che, però, mal si adatta ai cambiamenti che inevitabilmente intervengono durante il processo di sviluppo. Scrum adotta un **approccio iterativo e incrementale** per ottimizzare la prevedibilità e gestire il cambiamento.

Il risultato è quello di ottenere prodotti di **altissimo valore** che, soprattutto nell'ambito di progetti complessi che possono durare parecchi mesi, tendono a modificarsi in funzione di nuove esigenze che non erano presenti all'inizio del progetto e si sono palesate a lavori avviati.

Scrum prevede che i team di lavoro si auto-organizzino per portare a termine il proprio lavoro senza che qualcuno che non fa parte del team li diriga: questa libertà intrinseca consente al team e ai suoi membri di attingere in maniera **creativa** alle risorse messe a disposizione dal **framework** per trovare la strada più **efficace** per il raggiungimento degli obiettivi. Quindi, l'obiettivo di Scrum è di avere un team composto da un **numero piccolo di persone** che impiega un **periodo breve di tempo** per **realizzare una cosa piccola** che andrà a costituire un tutto, anziché un gruppo di persone numerosi che realizza un obiettivo grande.

---

## Le raccomandazioni di Scrum

Come abbiamo detto in precedenza Scrum non impone un metodo, ma suggerisce un framework costituito da pratiche, ruoli, eventi e strumenti per raggiungere gli obiettivi progettuali.

Di seguito riportiamo una veloce carrellata del framework, premettendo che solo una parte di esso compare, impiegato per gli scopi didattici che SMART CODING si prefigge, nella proposta di attività presente nella Guida.

Per chi volesse approfondire si rimanda a <http://goo.gl/zyraso>.

## Pratiche:

- **Suddivisione in team:** suddividere il gruppo di lavoro in team che si organizzano autonomamente e collaborano tra di loro.
- **Divisione del lavoro:** suddividere il problema complesso in una lista di problemi più semplici e per ognuno di questi individuare una soluzione che sia realizzabile e concreta (*deliverable*). Quindi ordinare l'elenco per priorità.
- **Organizzazione il lavoro per iterazioni:** dividere il tempo complessivo da dedicare al progetto in iterazioni di durata definita e per ciascuna individuare l'elenco degli oggetti da rilasciare, per esempio in base alla priorità definita al punto precedente.
- **Ottimizzazione del piano di rilascio:** revisione delle priorità e delle singole attività sulla base di quanto prodotto al termine di ogni iterazione.
- **Ottimizzazione del processo:** guardarsi indietro dopo ciascuna iterazione e identificare con i membri del team ciò che ha funzionato bene e cosa deve essere rivisto in un'ottica di miglioramento continuo.

## Ruoli:

- **Product owner:** è colui che conosce esattamente ciò che si desidera realizzare e ha il compito di garantire che il risultato raggiunto sia davvero quello atteso.
- **Team di sviluppo:** è formato da coloro che hanno il compito di tradurre le indicazioni del *product owner* in oggetti finiti che concorreranno al raggiungimento dell'obiettivo finale.
- **Scrum master:** ha il compito di assicurarsi che le pratiche suggerite da Scrum siano comprese e messe in atto da tutti gli attori. È un facilitatore.

## Eventi:

- **Sprint:** è la singola iterazione che si conclude con una dimostrazione degli oggetti rilasciati da parte del *Team al Product Owner* in un'ottica di revisione e aggiornamento del piano di rilascio.
- **Sprint planning:** è la pianificazione delle attività che dovranno essere effettuate dal team durante lo sprint.
- **Daily meeting:** è una micro-riunione di durata prefissata in non più di 5/10 minuti (*time boxed*), che precede o segue le attività giornaliere, dove i membri del team si confrontano e si scambiano opinioni e aiuti sul lavoro che hanno concluso o si apprestano a iniziare.
- **Sprint review:** è una riunione strutturata che si tiene al termine dello *Sprint* con l'obiettivo di rivedere il lavoro svolto per evidenziare errori, pratiche virtuose emerse durante le attività, aspetti da migliorare in vista dello *Sprint* successivo.

## Strumenti:

- **Stories:** è la traduzione dei singoli requisiti in attività concrete realizzata dai membri del team.
- **Product backlog:** è l'elenco di tutti i requisiti necessari al raggiungimento dell'obiettivo finale. È gestito e aggiornato dal *Product Owner*.
- **Sprint backlog:** è l'elenco dei soli requisiti che verranno presi in considerazione durante la singola iterazione (*Sprint*) e delle possibili soluzioni per soddisfarli (*Story*). È gestito e aggiornato dal Team.
- **Scrum board:** visualizza il piano e l'avanzamento dei lavori per il singolo *Sprint*
- **Burndown chart:** mette in evidenza attraverso un *planning* giornaliero se il team sarà in grado di produrre quanto pianificato per quello *Sprint* nei tempi previsti.

## Scrum e il Pensiero Computazionale

Si possono sottolineare una serie di analogie e assonanze:

- **Collaborazione:** il Pensiero Computazionale e Scrum esaltano l'importanza delle persone, della comunicazione tra di esse e del lavoro di gruppo, come strumenti per raggiungere obiettivi più grandi di quelli che il singolo potrebbe raggiungere da solo.
- **Approccio iterativo e incrementale:** sia il Pensiero Computazionale sia Scrum riconoscono l'importanza della suddivisione di un problema complesso in problemi più semplici, così che la soluzione di questi ultimi consenta, attraverso un approccio per tentativi ed errori (approccio empirico), di arrivare alla soluzione del problema più grande.
- **Creatività:** il Pensiero Computazionale e Scrum riconoscono il ruolo della creatività come veicolo di espressione di se stessi e di propensione al miglioramento continuo.