

## Framework

Methodology (Cambridge dictionary):

“ a **system** of **ways** of doing, **teaching**, or **studying** something”. In other words, it means that a methodology is a **systematic way of doing something**.

Framework:

a guideline or frame that we can work under.

So let's put this understanding in the context of project management. **A methodology is much more prescriptive**, it should tell us what steps to take, in what order and how to perform those steps but, most importantly, the reasons 'why' those steps should be taken, in particular order.”

Disadvantage of methodology is the limitation it has for any innovation or creativity since you need to follow pre-defined steps and procedures but at the same time it provides consistency

On the other hand, a framework allows you to be loose and flexible; it provides enough room for creativity. But at the same time, it has some level of ambiguity inherited in it, because in framework, you are not guided through specific steps or process to produce a certain result instead you are left to find your own way through.

(<https://www.linkedin.com/pulse/what-difference-between-methodology-framework-dafir-jacob#:~:text=I%20found%20a%20better%20definition,systematic%20way%20of%20doing%20something.&text=Oxford%20defines%20framework%20as%20%E2%80%9CA,%2C%20concept%2C%20or%20text%E2%80%9D.>)

Scrum è volutamente incompleto ed è fondato sull'intelligenza collettiva delle persone che lo adottano. Mette a disposizione una serie di strumenti che permettono di agevolare l'adattamento. Permette di essere adattato alle esigenze man mano che queste si presentano.

Nella realtà aziendale, Scrum non è mai la pedissequa applicazione meccanica di quanto scritto nella *Scrum Guide*. E' un riferimento

## Principi

### Controllo empirico del processo

La conoscenza si basa sull'esperienza. Si decide in base a quello che si osserva.

Si prova, si fanno esperimenti e si corregge se serve

Si basa su questi pilastri:

- **Trasparenza:** Il processo e il lavoro devono essere visibili a tutti. Le decisioni si prendono in modo corretto quando si ha accesso al maggior numero di informazioni possibile.  
⇒ La trasparenza permette l'ispezione
- **Ispezione** (attraverso: scrumboard, frequent feedback, Epic, Product Backlog, Release Planning, Dimostrazione dell'avanzamento). Tutti gli *artifact* devono essere ispezionati frequentemente  
⇒ L'ispezione permette l'adattamento
- **Adattamento** (attraverso: Daily standup Meeting, Identificazione dei rischi, change requests, Retrospect sprint Meetings ). Apprendendo qualcosa attraverso l'ispezione si possono mettere in pratica tutti gli accorgimenti che servono a correggere gli errori.  
⇒ Permette il miglioramento continuo

## Self organization

Auto-organizzazione non significa che i membri del team possono agire in qualunque modo vogliano. Significa solo che gli Scrum Team “decidono al loro interno chi fa cosa, quando e come” e non solo genericamente il modo migliore per completare il loro lavoro.

Ciascun membro del team è responsabile di stabilire quale lavoro lui o lei eseguirà.

Benefici:

- Adesione totale (c.d. buy-in) del team e responsabilità condivisa
- Motivazione, che porta ad un miglioramento del livello della prestazione del team
- Ambiente innovativo e creativo che contribuisce alla crescita

Obiettivi:

- Comprendere la Vision del Progetto e perché il progetto realizza valore per l'organizzazione
- Stimare le User stories e assegnarsele
- Identificare le attività in modo indipendente durante il processo *Identificare le Attività*
- Applicare e sfruttare la propria competenza di team interfunzionale per lavorare sulle attività durante il processo *Creare i Deliverable*
- Consegnare risultati tangibili
- Risolvere insieme i problemi individuali
- Chiarire eventuali discrepanze o dubbi ed essere aperti ad apprendere nuove cose
- Aumentare continuamente la conoscenza e la capacità con regolari interazioni all'interno del team

Mantenere la stabilità dei membri del team per tutta la durata del progetto, non cambiandone i membri a meno che non sia inevitabile.

## Collaboration

Cooperazione: quando il lavoro prodotto è composto dalla somma dei lavori delle singole persone

Collaborazione: quando un team lavora insieme per trarre vantaggio dai reciproci contributi per produrre qualcosa di più grande

Awareness: ognuno ha la consapevolezza del lavoro degli altri

Articulation: divisione del lavoro in parti più piccole

Appropriation: adattare la tecnologia alla situazione

Benefici:

- I rischi identificati e trattati in modo efficiente
- Riduzione al minimo dei cambiamenti dovuti a requisiti mal chiariti
- Realizzazione del pieno potenziale di un team
- Miglioramento continuo

## Time boxing

Benefici:

- Processi efficienti di sviluppo
- Minori spese generali
- Alta velocità dei team

Time box:

- Gli eventi di Scrum sono time-boxed
- Sprint: da 1 a 6 settimane (Consigliato da 4 settimane)
- Daily Standup Meeting: 15 minuti
- Sprint planning Meeting: 8 ore per uno sprint di un mese
- Sprint Review Meeting: 4 ore per uno Sprint di un mese
- Sprint Retrospect Meeting: 4 ore per uno Sprint di un mese

### **Value based Prioritization**

Durante la prioritizzazione devono essere tenuti in conto questi fattori: Valore, Rischio, Dipendenze

### **Iterative development**

Iterazione: ripetizione di un processo allo scopo di avvicinarsi all'obiettivo

Vantaggio: permette la correzione dell'andamento

## **Valori di un Team Scrum**

Impegno, Focus, Apertura, Rispetto e Coraggio

### **Aspetti**

### **Ruoli (responsabilità)**

Responsability vs accountability

Scrum team: numero ideale di persone da 6 a 10

Team cross-functional (o team interfunzionale): composto da persone con che arrivano da aree funzionali diverse e con prospettive diverse e nel complesso hanno tutte le competenze necessarie per creare valore allo sprint.

Autogestione già detta

Non ci sono gerarchie

### **Developers:**

Hanno la responsabilità di creare i deliverable attraverso successivi increment

Ogni increment deve rispettare la Definition Of Done ed è responsabilità del developer

Hanno la responsabilità di stimare le attività all'interno dello sprint backlog

Adattare la pianificazione all'interno dello sprint backlog

### **Product Owner:**

Il Product Owner deve sempre mantenere un duplice punto di vista. Deve capire e supportare i bisogni e gli interessi di tutti gli stakeholder e nel contempo comprendere anche le esigenze ed il funzionamento dello Scrum Team.

Responsabilità verso gli stakeholder:

Il Product Owner rappresenta gli interessi della comunità degli stakeholder di fronte allo Scrum Team.

Ha la responsabilità di portare valore agli stakeholder

Responsabilità verso il Team:

Il Product Owner ha la responsabilità di fare in modo che i requisiti di funzionalità del prodotto o servizio siano comunicati in modo chiaro allo Scrum Team, di definire i Criteri di Accettazione e di assicurarne la soddisfazione. In altre parole, il Product Owner deve assicurare la consegna di valore da parte dello Scrum Team.

Il Product Owner ha la responsabilità di tenere vivo l'impegno degli stakeholder. Le azioni che aiutano a farlo sono:

- ❖ Assicurare l'effettiva collaborazione e la partecipazione degli stakeholder al progetto
- ❖ Valutare continuamente l'impatto sul business
- ❖ Mantenere comunicazioni regolari con gli stakeholder
- ❖ Gestire le aspettative degli stakeholder

#### Responsabilità:

- ❖ Crea i requisiti complessivi iniziali del progetto e fa partire il progetto
- ❖ Nomina le persone appropriate per i ruoli di Scrum Master e Scrum Team
- ❖ Procura le risorse finanziarie iniziali e correnti per il progetto
- ❖ Stabilisce la Vision del Prodotto
- ❖ Valuta la fattibilità ed assicura la consegna del prodotto o servizio
- ❖ Assicura la trasparenza e la chiarezza degli elementi del Prioritized Product Backlog
- ❖ Decide il contenuto minimo di un rilascio commerciabile
- ❖ Fornisce i Criteri di Accettazione delle User Story da sviluppare in uno Sprint
- ❖ Esamina i deliverable
- ❖ Decide la durata dello Sprint
- ❖ Consegna valore con i progetti
- ❖ Crea la giustificazione commerciale dei progetti
- ❖ Conferma la realizzazione dei benefici agli Stakeholder

Uno delle responsabilità del Product Owner è la scrittura delle User Story. In genere il PO scrive le US da solo, anche se in alcuni casi si può avvalere di un workshop sulla scrittura delle User Story.

#### **USER Story**

##### *Formato della User Story:*

Nella mia qualità di <ruolo/persona>, dovrei essere in grado di <requisito> in modo da <beneficio>.

##### *Esempio di User Story:*

Nella mia qualità di Amministratore del Database, dovrei essere in grado di tornare indietro di un determinato numero di aggiornamenti del database in modo da poter ripristinare la versione del database desiderata.

Anche se il Product Owner approva le User Story iniziali di uno Sprint, la decisione finale su quali specifiche User Story (fra quelle approvate) scegliere per lo Sprint spetta allo Scrum Team. Lo Scrum Team (di concerto con il Product Owner, se necessario) definisce a quali User Story lavorerà durante lo Sprint (FZ: dato che è lo Scrum Team che ha la responsabilità di consegnarle per una data prestabilita).

#### **Scrum master:**

Non è il capo

Deve promuovere scrum all'interno del team

È focalizzato su come far crescere il team e su come migliorare ad esempio su quali esperimenti provare

È responsabile dell'efficacia del team

È il leader al servizio del team

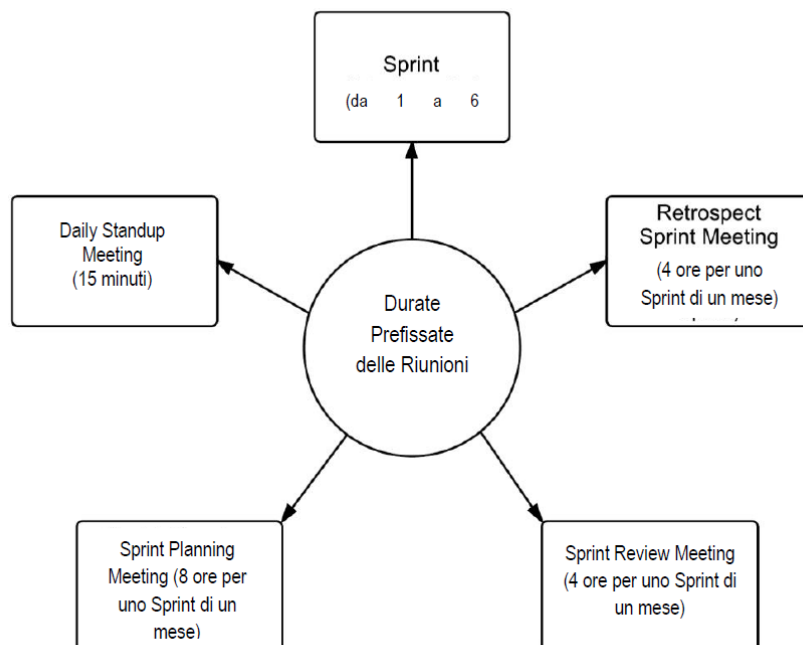
Allena i membri del team all'autogestione e alla cross funzionalità

Si assicura che tutti gli eventi avvengano nella modalità corretta

Fa da facilitatore sia con il PO che eventualmente con gli Stakeholder

Aiuta nella pianificazione

## Eventi



### Release planning meeting

In Scrum, l'obiettivo principale di un Release Planning Meeting è quello di permettere allo Scrum Team di avere una visione d'insieme dei rilasci e del programma di consegna del prodotto che il team sta sviluppando, in modo da potersi allineare alle aspettative del Product Owner e degli stakeholder pertinenti (primo fra tutti lo sponsor del progetto).

### Rilascio continuo vs rilascio per fasi

Molte organizzazioni hanno una strategia riguardo al rilascio dei prodotti. Alcune preferiscono la distribuzione continua, nella quale si fa un rilascio dopo la creazione di una specifica funzionalità utilizzabile. Altre preferiscono la distribuzione per fasi, nella quale i rilasci vengono fatti a intervalli predefiniti. A seconda della strategia adottata dall'organizzazione, le Sessioni di Pianificazione dei Rilasci nei progetti possono essere basate sulle funzionalità, nel qual caso l'obiettivo è consegnare un rilascio ogni volta che viene sviluppato un gruppo prestabilito di funzionalità; oppure la pianificazione può essere basata sulle date, nel qual caso il rilascio avviene in date prestabilite.

Dal momento che il framework Scrum promuove il processo decisionale iterativo e basato sulle informazioni raccolte piuttosto che la pianificazione anticipata di dettaglio praticata nello stile tradizionale di project management a cascata, **le Sessioni di Pianificazione dei Rilasci non devono produrre un Piano dei Rilasci dettagliato per l'intero progetto**. Tale piano può essere aggiornato continuamente man mano che sono disponibili informazioni rilevanti.

Non è detto che vi sia un rilascio programmato alla fine di ogni Sprint. A volte, un rilascio può essere previsto dopo il completamento di un gruppo di Sprint.

### **Product Backlog Meeting**

Il Product Owner lavora con il cliente per ricevere i requisiti di business e tradurli in User Story fattibili e per capire quale di queste offrano il maggior valore. Allo stesso modo il PO deve lavorare con lo Scrum Team per capire i rischi, le incertezze e per capire le dipendenze tra le varie User Story.

Nel mettere in ordine le User Story all'interno del **Prioritized Product Backlog** occorre tenere in conto i seguenti fattori:

- ❖ Valore
- ❖ Rischio o incertezza
- ❖ Dipendenze

### **Sprint Planning Meeting**

**Sprint:** è il cuore di tutto il processo. È l'insieme di tutte le attività che servono a raggiungere uno scopo.

Scrum è un susseguirsi di sprint continuo e iterativo verso l'obiettivo finale.

Durante uno sprint ci sono una serie di eventi che servono per promuovere i principi di Scrum

Lo sprint ha uno scopo, definito all'inizio di uno sprint

Durata da una a 6 settimane. Se si finisce prima, si può decidere se mettere dentro qualcos'altro oppure far finire lo sprint

Se lo scopo viene meno, lo sprint si annulla

Perfezionamento del product backlog in base alle necessità

Definition of Done

Sprint più brevi aiutano a ridurre i costi.

Vediamo poi come mantenere aggiornato l'avanzamento (burn down e burn up ecc)

### **Sprint Planning Meeting**

Lo Sprint Planning è un meeting in cui il team pianifica il lavoro che deve essere svolto e portato a termine durante lo Sprint. A differenza del Refinement, in cui si ragiona a livello di insieme (user story), lo Sprint Planning prevede un maggior livello di dettaglio e una suddivisione in task. Il risultato dello Sprint Planning è lo **Sprint Backlog** ovvero l'elenco di attività che il team prevede di portare a termine entro la conclusione dello Sprint.

Fase 1 e fase 2. La prima con il PO, la seconda senza

Scomposizione delle User Story in task e relativa stima (Task planning meeting e Task estimation meeting).

8 ore per un mese

Criticità tra PO e Development team, su cosa mettere nello sprint. L'importanza dello Sprint Goal

### **Daily Scrum Meeting**

Ha lo scopo di tenere ispezionato l'avanzamento verso lo Sprint Goal.

Le tre domande, diventate ormai obsolete

Miglioramento delle comunicazioni

Si evidenziano i problemi

Impedimenti per lo scrum master da risolvere. Impediment backlog

Riducono la necessità di meeting (e soprattutto riducono la necessità di interruzioni)

Meeting successivi, più focalizzati e con un numero ridotto di persone

Anche se ci sono report, non va visto come uno strumento per fare report, ma va visto come un momento per aumentare la comunicazione e fare emergere i problemi il più presto possibile

15 minuti alla stessa ora

Partecipa tutto il team (PO opzionale)

### **Sprint Review Meeting**

Il meeting comincia con una dimostrazione di ciò che è stato fatto durante lo Sprint appena terminato. Si parte con la prima user story e si procede con quelle successive, fino a quando non si esauriscono i requisiti presenti nello Sprint Backlog.

Si mostra quello che è stato fatto nello sprint

Coinvolti gli stakeholder

Riadattamenti del product backlog

### **Durata**

4 ore per uno sprint di un mese.

### **Partecipanti**

Convocata dallo Scrum Team, partecipano lo Scrum Team, Product Owner, ScrumMaster, altri soggetti interessati se opportuno, se invitati dal Product Owner

### **Retrospect Meeting**

In Scrum il **Retrospective meeting** è l'occasione per il team di analizzare, in retrospettiva, lo Sprint appena concluso. Come dice il nome stesso, il meeting ha luogo al termine di un'iterazione, di solito al termine del Review meeting.

Anche il feedback degli utenti e degli stakeholder è importante e può essere portato al tavolo di discussione.

Ogni **Retrospective meeting** si basa su queste tre domande

1. Cosa è andato bene durante lo Sprint precedente? ()
2. Cosa possiamo migliorare?
3. Cosa ci impegnamo a migliorare nel prossimo Sprint?

I temi su cui si può riflettere sono molteplici, per esempio:

- ❖ Produttività
- ❖ Processo
- ❖ Risultati
- ❖ Team
- ❖ Strumenti di lavoro
- ❖ Ambiente di lavoro

Il Registro della Retrospettiva dello Sprint è una registrazione delle opinioni, delle discussioni e delle iniziative perseguibili emerse in un Retrospect Sprint Meeting.

L'insieme di tutti i Registri delle Retrospective degli Sprint diventa il diario di progetto e dettaglia i successi, le questioni, i problemi del progetto e le soluzioni. I registri sono documenti pubblici a disposizione di tutti coloro che fanno parte dell'organizzazione.

I Miglioramenti Fattibili Concordati sono il principale output del processo Retrospettiva dello Sprint e rappresentano l'elenco delle iniziative perseguibili messe a punto dal team per affrontare i problemi e migliorare i processi al fine di aumentare le proprie prestazioni negli Sprint a venire.

Di solito i miglioramenti più importanti vengono inclusi come items nello sprint successivo.

Lo Scrum Master in tal caso non dovrebbe esprimere un giudizio sulla bontà o sulla utilità di tali azioni ma dovrebbe invece stimolare il gruppo a rispondere a domande del tipo:

- Rispetto agli obiettivi che ci siamo dati (p.e. in che modo lavorare come team o in che modo realizzare il prodotto secondo le indicazioni del PO) quanto e come questa azione potrebbe aiutarci ad avvicinarci a tali obiettivi?
- Rispetto alle metriche di performance (p.e. tempo di chiusura di un bug, time to delivery/deploy), in che modo questa azione potrebbe permetterci di ottenere una prestazione migliore?
- E se questa azione risponde ai punti precedenti, come potremmo dire di averla fatta/conclusa?
- Se invece si tratta di un esperimento — per definizione difficilmente collegabile alle metriche di cui sopra, perché un esperimento è innovazione — come possiamo renderci conto che abbiamo avuto successo?
- E in caso di successo come potremo propagare? O, in caso di fallimento, come potremo limitare l'espansione del danno?

**Durata**



4 ore per uno sprint di un mese

## Partecipanti

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

## Artefatti

Product backlog:

Deve rispettare il Product Goal (è l'obiettivo a lungo termine del prodotto)

Elenco di attività priorizzate

Sprint backlog:

deve rispettare lo Sprint Goal (obiettivo dello sprint)

Elenco delle user story che si prendono dal product backlog e che si competeranno all'interno dello sprint. Viene aggiornato durante lo sprint

Increment:

deve rispettare la Definition Of Done

Si aggiunge all'increment precedente e di increment in increment si raggiunge il risultato finale del progetto. In caso contrario, non sarà un increment, non viene rilasciato e non viene nemmeno presentato all'interno della Sprint Review

Un elemento del product backlog, nel momento in cui soddisfa la Definition Of Done, costituisce un increment

Definition of done

Stabilisce che cosa vuol dire completare una cosa

La definition of done è generica per tutte le US. I criteri di accettazione sono specifici. I criteri di accettazione sono requisiti tecnici e/o funzionali. La DoD è un requisito di progetto

Esempio di Definition Of Done

- Scrivere Unit Test
- Code review
- Codice presente su VCS
- Eseguire il deploy sull'ambiente di stage
- Eseguire test di accettazione