



**Corso di Laurea magistrale
in Management della
transizione digitale**

Design della trasformazione digitale

**Sante Dotto
Vittorio D'Orsi**

A.A. 2024-2025



Project management e innovazione

Caso di studio: Agile Scrum Framework

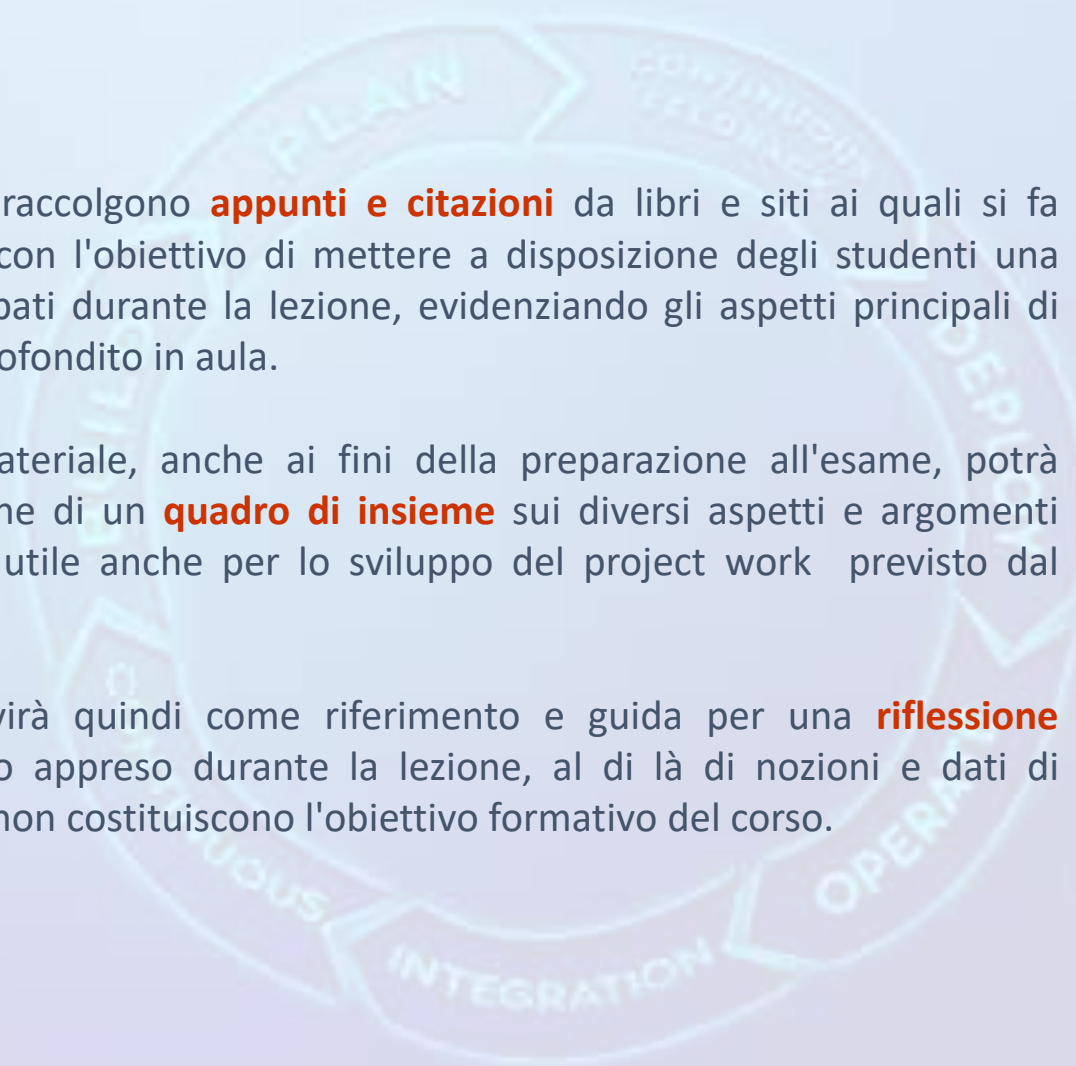
Sante Dotto

Testi e immagini tratti da:

<https://www.zerounoweb.it/cio-innovation/metodologie/agile-transformation-come-cambia-il-modo-di-lavorare-per-affrontare-la-trasformazione-digitale/>

www.slideshare.net/stefanovalle/introduzione-alle-metodologie-di-sviluppo-agile

Foto di Dirk Wouters da Pixabay



Le slide che seguono raccolgono **appunti e citazioni** da libri e siti ai quali si fa riferimento nel testo, con l'obiettivo di mettere a disposizione degli studenti una traccia dei temi sviluppati durante la lezione, evidenziando gli aspetti principali di quanto discusso e approfondito in aula.

L'utilizzo di questo materiale, anche ai fini della preparazione all'esame, potrà facilitare la ricostruzione di un **quadro di insieme** sui diversi aspetti e argomenti oggetto della lezione, utile anche per lo sviluppo del project work previsto dal corso.

Questo materiale servirà quindi come riferimento e guida per una **riflessione complessiva** su quanto appreso durante la lezione, al di là di nozioni e dati di dettaglio che di per sé non costituiscono l'obiettivo formativo del corso.

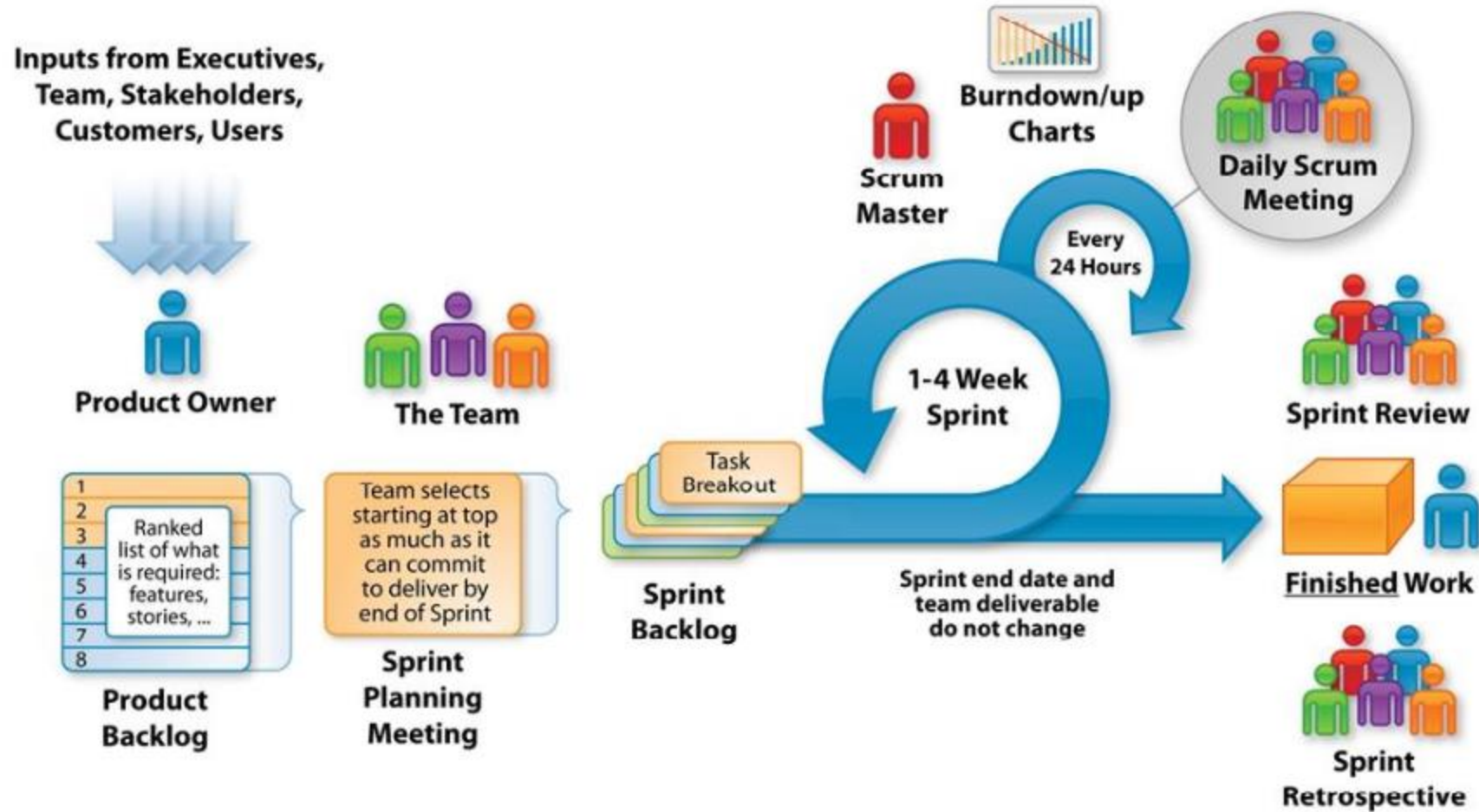
AGILE SCRUM FRAMEWORK

Nato alla fine degli anni '80 dall'esperienza di Ken Schwaber e Jeff Sutherland, due dei firmatari del Manifesto Agile, è il framework di gran lunga più diffuso in ambito trasformazione digitale.



Prende il nome dalle mischie del rugby, dove tutti gli uomini spingono insieme nella stessa direzione. Questo è il principio alla base di SCRUM: fare in modo che **tutti i membri del team** lavorino coordinati verso il medesimo obiettivo.

The Agile - Scrum Framework



SCRUM è caratterizzato dai seguenti elementi:

- focus sul lavoro all'interno di team dedicati a **specifici obiettivi** di sviluppo applicativo, composti da non più di 10 persone
- i **ruoli** sono:
 - Product Owner, definisce obiettivi e priorità
 - SCRUM Master, facilita il lavoro dei membri del team
 - Componenti del team
- lo sviluppo del **software** è gestito tramite un «*product backlog*», una lista prioritizzata di funzionalità (chiamate «*storie*»)
- il lavoro è organizzato per **iterazioni** chiamate «*sprint*» della durata di 2-4 settimane durante le quali si realizzano le storie a maggiore priorità
- esistono diversi **eventi** per permettere al team di lavorare bene insieme:
 - brevi «*stand up meeting*» giornalieri per il coordinamento continuo
 - «*sprint planning meeting*» per definire cosa fare in uno sprint
 - «*sprint review*» per condividere i risultati del lavoro realizzato all'interno di uno sprint
 - «*retrospective*» per riflettere sulle modalità di lavoro e migliorarle
- l'**analisi di dettaglio** è fatta all'ultimo momento possibile per permettere di raccogliere maggiori informazioni e sono introdotte forme collaborative di analisi e progettazione improntate ad ottenere una piena comprensione e condivisione di funzionalità e priorità, come ad esempio lo «*user story mapping*».

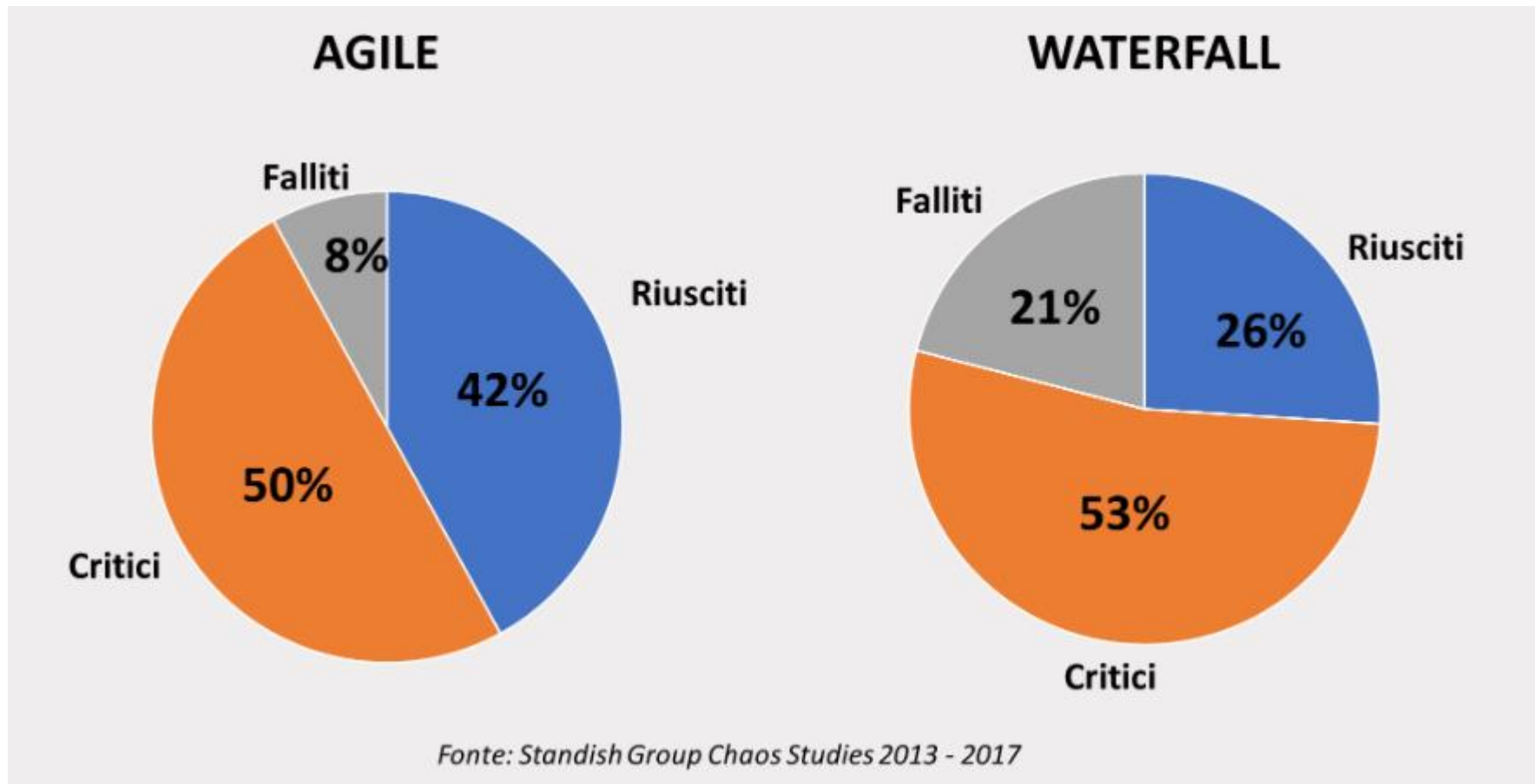
Lavorando in questo modo il team è in grado di accogliere a valle di ogni *sprint* i *feedback* da parte dei diversi stakeholder e riprogrammare di conseguenza le proprie attività, garantendo quindi massima **flessibilità**. Le persone sono responsabili delle storie assegnate e si coordinano frequentemente per evitare asimmetrie informative e per migliorare di continuo.

Questo approccio permette di definire inoltre molto chiaramente le **priorità** di un prodotto - evitando di realizzare funzionalità non utili - e di inserire metriche precise per verificare l'andamento del progetto e dare massimo controllo su quello che viene realizzato.

In quest'ottica, siccome lo *scope* del progetto può variare sulla base del cambiamento del contesto, è possibile fissare inizialmente tempi e costi: il team ha l'obiettivo di massimizzare il **valore prodotto** all'interno di questi vincoli, evitando di realizzare funzionalità non utili.

SCRUM può essere utilizzato anche da **unità più ampie**, oltre che essere applicato ai singoli team di sviluppo, coordinando il lavoro di diversi team all'interno della stessa organizzazione. Per questo obiettivo esistono diverse declinazioni del framework: le più note sono SAFe (Scaled Agile Framework), Less (Large Scale SCRUM), Scrum of Scrum, DaD (Disciplined Agile Delivery) e Nexus.

L'utilizzo di SCRUM si è dimostrato estremamente utile nel corso degli anni e ha prodotto significativi **risultati sull'andamento dei progetti** (Standish Group Chaos Studies 2013 - 2017).



PROBLEMATICHE DA AFFRONTARE 1/2

- **Competenze** - occorre integrare e sviluppare i nuovi ruoli ed evitare la tendenza a sviluppare competenze tecniche specialistiche, cosa che ostacola la collaborazione, prediligendo uno spettro di competenze più ampio
- **Resistenza al cambiamento** - rimuovere gli ostacoli culturali che le persone tendono a manifestare dato l'elevato impatto di queste nuove modalità anche adottando, ad esempio, nuovi processi di *performance management* orientati allo sviluppo ed alla responsabilizzazione.



PROBLEMATICHE DA AFFRONTARE 2/2

- **Procurement** - nuove forme contrattuali per instaurare con i fornitori relazioni di partnership
- **Organizzazione** - la centralità dei team si scontra con i silos organizzativi preesistenti, andando a mettere in discussione l'utilità di tutto il middle management
- **Business Agility** - creazione di team cross-funzionali dedicati a obiettivi specifici che sappiano anteporre gli obiettivi e le priorità del progetto a quelli della propria funzione
- **Coordinamento** - orientato al lungo periodo e alla crescita delle persone nell'organizzazione, piuttosto che allo scambio di know-how → famiglie professionali, comunità di pratica e altre forme leggere di coordinamento

