Corso UFT06 - Project Management

Pinna Pier Nicolò

**Glossario**

**SLIDE L1**

**Sommario**

Il Project Manager

Chi è

Il triangolo del project management

Progetto vs Processo

Il Project Management nella pratica

Cosa significa pianificare

Tipologie di progetto & project management

Predittivo vs Adattivo

**Il project manager**

Il project manager è responsabile della pianificazione e controllo di progetti, dall’idea sino al loro completamento. Coordina persone e processi con l’obiettivo di consegnare progetti nel tempo, budget e nei requisiti richiesti per raggiungere uno o più obiettivi.

**Il triangolo del project management**

Tempo, Costo, Scopo = Qualità

Il triangolo non può crescere o rimpicciolirsi, ma può solo variare.

La variazione positiva o negativa di una delle tre voci, porterà una conseguente variazione positiva o negativa di una o entrambe le altre voci.

Esempio: Se voglio fare di più, spenderò di più, oppure avrò bisogno di più tempo.

E’ impossibile fare di più, spendendo uguale nello stesso tempo.

**Progetto**

Uno sforzo temporaneo intrapreso per creare un servizio o un risultato di progetto unico, in un lasso di tempo definito.

**Processo**

Insieme di attività eseguite ripetutamente, con lo scopo di ottenere lo stesso risultato.

**Pianificare**

La pianificazione è il processo di pensare alle attività richieste per il raggiungimento degli obiettivi desiderati.

**Modelli di project management**

* Predittivo
* Adattivo

**Predittivo**

Un progetto in grado di consentire anticipazioni e previsioni:

* Guidato da un piano
* Requisiti fissi
* Analisi utente più importante della scoperta
* Grande sforzo di Design all’inizio
* Architettura predefinita
* Stime fornite all’inizio
* Previsioni basate sulle stime iniziali
* Tutti i requisiti hanno la stessa priorità

**Adattivo**

Un progetto che favorisce o consente l'adattamento:

* Guidato dalla pianificazione
* Requisiti variabili
* La scoperta è più importante dell’analisi
* Sforzo minimo indispensabile di Design all’inizio
* Architettura si evolve
* Stime in costante aggiornamento
* Previsioni basate sul lavoro già svolto
* Requisiti prioritizzati a seconda del loro valore per l’utente

**Metodologie di project management:**

* Waterfall
* Agile

**Waterfall**

Il modello a cascata è il più tradizionale modello di ciclo di vita del software. Secondo questo modello, il processo di realizzazione del software è strutturato in una sequenza lineare di fasi o passi

**Il rischio nel waterfall**

Non viene ridotto significativamente sino alla fine del progetto

**Stato di progetto rispetto all’esigenza nel waterfall**

Lo stato di progetto tende ad allontanarsi dall’esigenza reale del business, perché questa varia nel tempo, mentre il progetto inizia e finisce con requisiti fissi e invariabili

**Agile**

I metodi agili, propongono un approccio focalizzato sull'obiettivo di consegnare al cliente, in tempi brevi e frequentemente, software funzionante e di qualità.

**Il rischio nell’ agile**

Si riduce significativamente dopo ogni iterazione

**Stato di progetto rispetto all’esigenza nell’ agile**

Lo stato di progetto si avvicina ad ogni iterazione all’esigenza reale del business, perché questa varia nel tempo, ed il progetto si adatta alle nuove esigenze

**SLIDE L2**

**Sommario**

Il manifesto agile

Kanban vs Scrum

Scrum

I ruoli

Le fasi

Le cerimonie

Il backlog Scrum

**Il manifesto agile**

Il manifesto per lo sviluppo agile di software è un documento redatto nel febbraio 2001 da diciassette sviluppatori con l’intento di definire i valori e i principi chiave dello sviluppo agile.

* **Our highest priority** is to satisfy the customer through early and continuous delivery of valuable software.
* **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
* **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
* **Business people and developers** must work together daily throughout the project.
* **Build projects around** motivated individuals.
* **Give them the environment and support they need**, and trust them to get the job done
* **The most efficient and effective method** of conveying information to and within development team is face-to-face conversation.
* Working software is the primary measure of progress.
* **Agile processes promote** sustainable development.
* **The sponsors, developers, and users** should be able to maintain a constant pace indefinitely.
* **Continuous attention** to technical excellence and good design enhances agility.
* **Simplicity** – the art of maximizing the amount of work not done – is essential.
* The best architectures, requirements, and designs emerge from self-organizing teams.
* **At regular intervals**, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

**Metodologie di progetto dell’agile:**

* Kanban
* Scrum

**Kanban**

* Un flusso continuo di attività
* Lo scope può variare, le attività vengono prese una alla volta a seconda del loro ordine nella board
* La velocità del team viene misurata con cadenza fissa sul quantitativo di oggetti completati
* Non ci sono ruoli predefiniti
* È facile adattarsi
* È reattivo verso i cambiamenti
* Ruoli fluidi e Project Manager opzionale
* I task sono condivisi da tutti
* La timeline si evolve a seconda delle esigenze
* I cambiamenti vengono gestiti in mezzo alla timeline di progetto
* La produttività è misurata sulla lunghezza dell’intero progetto

**Scrum**

* Brevi iterazioni (1 o 2 settimane)
* Lo scope è definito per ogni iterazione
* La velocità del team viene misurata sulla base di risultati precedenti
* Un insieme di attività definite per ogni iterazione
* Ci sono ruoli predefiniti
* È difficoltoso adattarsi
* Non reattivo verso i cambiamenti
* Ruoli predefiniti e Scrum Master obbligatorio
* I task sono assegnati a specifici responsabili
* È difficoltoso adattarsi
* La timeline è fissa e definita in Sprint
* I cambiamenti vengono gestiti alla fine di ogni sprint
* La produttività viene misurata sulla velocità in Story point alla fine di ogni sprint

|  | Kanban | Scrum |
| --- | --- | --- |
| Attività | Un flusso continuo di attività | Brevi iterazioni (1 o 2 settimane)  Un insieme di attività definite per ogni iterazione |
| Scope | Lo scope può variare, le attività vengono prese una alla volta a seconda del loro ordine nella board | Lo scope è definito per ogni iterazione |
| Velocità | La velocità del team viene misurata con cadenza fissa sul quantitativo di oggetti completati | La velocità del team viene misurata sulla base di risultati precedenti |
| Ruoli | Non ci sono ruoli predefiniti | Ci sono ruoli predefiniti |
| Adattamento | È facile adattarsi | È difficoltoso adattarsi |
| Approccio ai cambiamenti | È reattivo verso i cambiamenti | Non reattivo verso i cambiamenti |
| Il project manager | Ruoli fluidi e Project Manager opzionale | Ruoli predefiniti e Scrum Master obbligatorio |
| I task | I task sono condivisi da tutti | I task sono assegnati a specifici responsabili |
| Timeline | La timeline si evolve a seconda delle esigenze | La timeline è fissa e definita in Sprint |
| Gestione cambiamenti | I cambiamenti vengono gestiti in mezzo alla timeline di progetto | I cambiamenti vengono gestiti alla fine di ogni sprint |
| Produttività | La produttività è misurata sulla lunghezza dell’intero progetto | La produttività viene misurata sulla velocità in Story point alla fine di ogni sprint |

**Ruoli nello scrum**

* Product owner
* Scrum master
* Developer

**Product owner**

* Responsabilità di massimizzare il ROI
* Definisce priorità
* Interagisce con i Developer
* Responsabilità sul lavoro svolto

**Scrum master**

* Aiuta il team a raggiungere gli obiettivi
* Protegge il team da ogni rumore esterno
* Istruisce il gruppo sull’uso di Scrum
* Facilita il processo

**Developer**

* Costruiscono il prodotto
* Sono un gruppo autogestito
* Si impegnano nel raggiungimento degli obiettivi
* Danno indicazione e suggerimenti al PO (product owner)

**Le fasi nell’ Agile Software Development del metodo Scrum**

* Requirements
* Plan
* Design
* Develop
* Release
* Track & Monitor / Retrospect

**Le cerimonie Scrum**

* Sprint planning
* Grooming/Refinement
* Estimation
* Daily standup
* Review
* Retrospective

**Sprint planning**

**Cos'è:** Lo Sprint planning è un evento che da origine allo sprint.

Lo scopo dello sprint planning è definire cosa può essere completato e consegnato nello sprint e come verrà svolto il lavoro necessario.

Lo sprint planning è generalmente svolto con la collaborazione dell’intero team.

**Durata:** 2h ogni settimana di sprint

Come condurre lo sprint planning:

The What – Il Product owner descrive gli elementi del planning e il team decide cosa aggiungere al prossimo sprint

The How – Il team di sviluppo decide come implementare le attività identificate. Successivamente c'è una sessione di confronto fra team di sviluppo e product owner rispetto alle priorità di business

The Who – Non è possibile fare lo sprint planning senza product owner o team di sviluppo. Il product owner definisce gli obiettivi di business. Il team di sviluppo capire come possono realizzare o meno gli obiettivi. Lo sprint planning non può essere svolto senza entrambe queste figure.

The Inputs – Lo strumento che si usa per guidare lo sprint planning è il backlog. Il team ha anche la possibilità di guardare sprint precedenti per migliorare eventuali stime.

The Outputs – La cosa più importante come output dello sprint planning è definire l’obiettivo dello sprint e come il team inizierà a lavorare per il raggiungimento dei suoi obiettivi.

**Grooming/Refinement**

**Cos'è:** Backlog grooming, anche definito backlog refinement o story time, è un evento ricorrente per team di sviluppo agile.

Lo scopo principale di questo meeting è assicurarsi di avere abbastanza story pronte per essere pianificate negli sprint successivi.

Durata: 2h ogni settimana di sprint

**Estimation**

**Cos'è:** L’estimation meeting un evento ricorrente per team di sviluppo agile.

Lo scopo principale di questo meeting è assegnare gli story points ad ogni attività.

Questa fase viene spesso inclusa durante lo sprint planning.

**Durata:** 1h ogni settimana di sprint

Tecniche più utilizzate per questo meeting

* Planning Poker
* Analogy
* T-Shirt Size
* Dot Voting
* The Bucket System
* Fibonacci series for Story Points

**Daily standup**

**Cos'è:** Il daily standup è un meeting giornaliero.

Lo scopo principale di questo meeting è fare un punto della situazione e capire eventuali criticità dello sprint

**Durata:** 15m ogni giorno

Il daily standup viene solitamente gestito con il seguente metodo:

* Cosa ho fatto ieri
* Cosa farò oggi
* Sono o non sono bloccato

**Sprint review**

**Cos'è:** Lo sprint review è un meeting che viene effettuato 1 volta durante tutto lo sprint.

Lo scopo principale di questo meeting è fare un punto della situazione e capire eventuali criticità dello sprint.

**Durata:** 15m ogni giorno

**Retrospective**

**Cos'è:** Il retrospective meeting è un appuntamento fisso per i team di sviluppo agile, generalmente svolto alla fine di uno sprint.

Lo scopo di questo meeting è analizzare come è andato lo sprint, capire eventuali problemi e identificare soluzioni per migliorare negli sprint successivi

**Durata:** 30m per ogni settimana di sprint, una volta a fine sprint

Esistono varie tecniche di retrospective:

* Start Stop Continue retrospective
* 4Ls retrospective
* Three Little Pigs retrospective
* Mountain Climber retrospective
* ecc…

**Il backlog in Scrum:**

Il Product Backlog è una lista ordinata di tutto ciò che deve essere, o vorresti che venisse, fatto per il tuo prodotto.

E' organizzata dal Product Owner, ma può essere riempita da chiunque (product team, clienti, stakeholders, scrum master, etc...)

**SLIDE L3**

**Sommario**

La pianificazione agile

Idea di alto livello

Identificazione Epic

WBS delle Epic in Story

Fornire le stime

Realizzare una roadmap

Condividere, revisionare, migliorare

**Pianificazione in agile**

1. Iniziare con una idea di alto livello
2. Identificare le macro attività (Epic)
3. Realizzare una WBS (Work Breakdown Structure) delle Epic in Story
4. Fornire le stime
5. Realizzare una roadmap
6. Condividere con il team e validare
7. Revisionare e migliorare
8. **Idea di alto livello**

Cosa stiamo sviluppando? Qual è l’obiettivo del progetto? Una casa? Il lancio di un nuovo prodotto? Una nuova campagna marketing? Un cambio di target per il proprio brand?

In un contesto aziendale Agile, ogni progetto potrebbe rientrare in un preciso “Tema”

**Tema**

Un tema è una grande area con elementi in comune, che aiuta i team Agile a tenere sotto controllo gli obiettivi aziendali.

1. **Identificare le macro attività (Epic)**

**Epic**

Un epic Agile è una porzione di lavoro che può essere suddivisa in task specifici (chiamati storie utente) in base alle esigenze/richieste dei clienti o degli utenti finali.

Esempio:

Progetto:

* Lancio di un nuovo prodotto

Epic:

* Analisi del mercato \*\*\*
* Definire il business
* Aggiornare le linee di produzione
* Far arrivare il prodotto negli store

**Definire le release**

Definire le release intermedie, consente di avere delle “Milestone” da inserire nella pianificazione, per supportare lo sviluppo e il raggiungimento di obiettivi.

Avere delle Milestone intermedie consente inoltre un maggiore controllo dello stato del progetto.

1. **Realizzare una WBS (Work Breakdown Structure) delle Epic in Story**

**User Story**

Una storia utente è una spiegazione informale e generale di una funzione del software scritta dalla prospettiva dell'utente finale. Il suo scopo consiste nello spiegare in che modo la funzione del software è in grado di fornire valore al cliente.

Esempio:

Story:

* Analisi del mercato \*\*\*
  + Come azienda, voglio sapere quanto è grande il mercato Asiatico del mio prodotto, per verificare il senso di lanciare il nuovo prodotto in Asia
  + Come azienda, voglio sapere quanto è grande il mercato Americano del mio prodotto, per verificare il senso di lanciare il nuovo prodotto in America

1. **Fornire le stime**

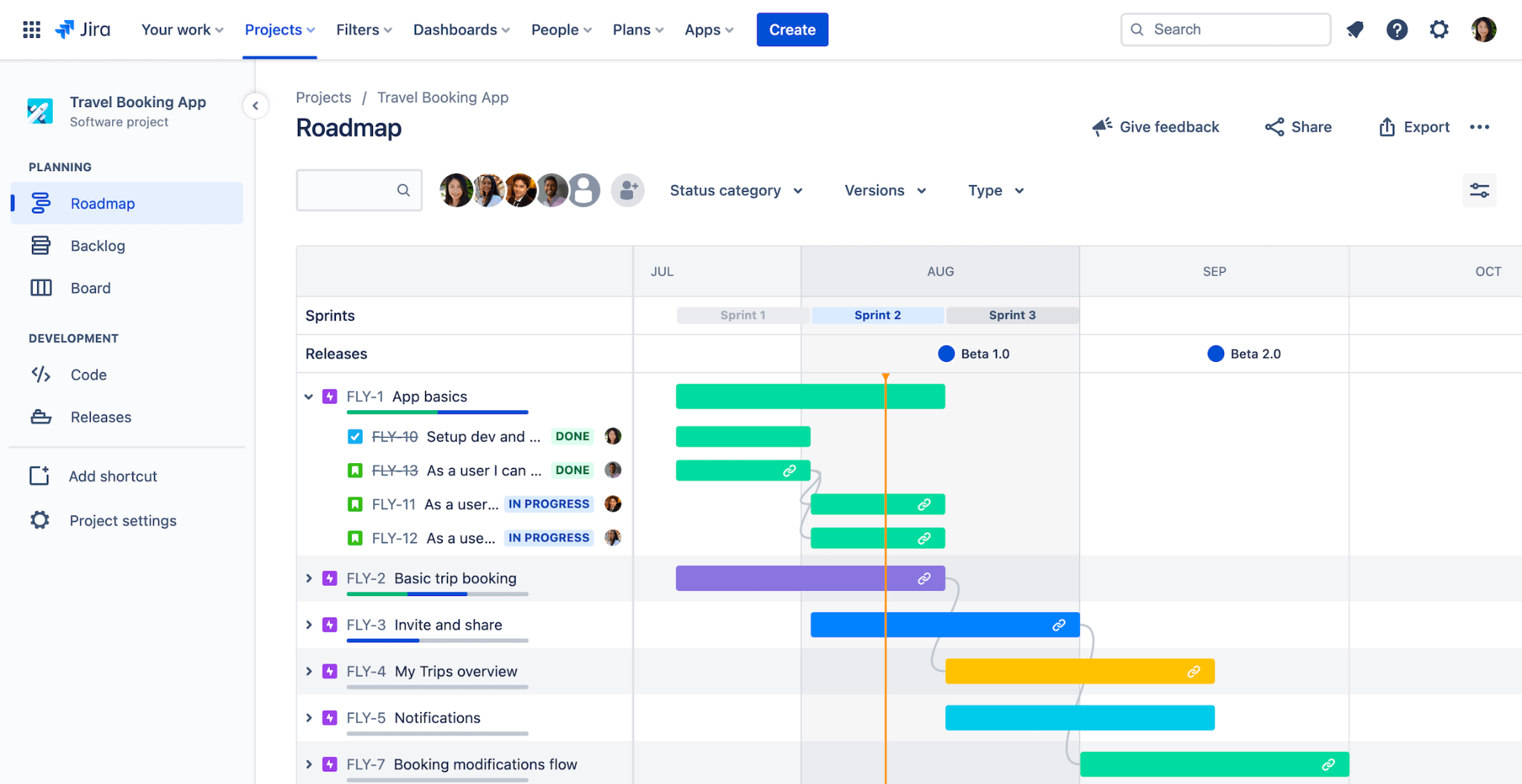
**Story points**

Story points are units of measure for expressing an estimate of the overall effort required to fully implement a item or any other piece of work.

1. **Realizzare una roadmap**

Il totale degli story point / la velocità del team = La data di fine progetto stimata

Esempi di roadmap:



NB:

* Scope di progetto definito in Tema/Iniziativa/Epic/Story
* Velocità del team
* Quante risorse abbiamo a disposizione nel nostro team

1. **Condividere, revisionare, migliorare**

In un team Agile, è fondamentale condividere i piani con il team di lavoro, per assicurarsi che la roadmap, il piano di progetto, lo sprint, siano sempre chiari e trasparenti verso il team stesso

**SLIDE L4**

**Sommario**

Come monitorare un progetto agile

Velocità pianificata vs velocità effettiva

How to calculate the team velocity average?

Che strumenti usare per il monitoraggio

Burndown & Burnup charts

Sprint report

Diagrammi cumulativi

I diagrammi di Gantt

**Monitorare un progetto agile**

Monitorare un progetto in corso di svolgimento è parte fondamentale del lavoro del Project Manager / Scrum Master ed è necessario al fine di prevenire o mitigare situazioni che possono recare danno al progetto.

**Elementi per il monitoraggio di un progetto agile:**

* Velocità pianificata del team vs velocità effettiva
* Data di fine progetto attesa, minima e massima
* Variazioni di costo minime e massime

**Velocità pianificata vs velocità effettiva**

Durante la pianificazione di un progetto scrum, identificheremo una velocità media del team.

Tuttavia, questa potrebbe variare durante gli sprint e avere un impatto sulla data di consegna del progetto.

Scenario:

1. Best case scenario

All sprints at the max team velocity

1. Worst case scenario

All sprints at the min team velocity

1. Most likely scenario

All sprints at the average team velocity

**How to calculate the team velocity average?**

Team Velocity = Total Story Points Completed Per Sprint / Number of Sprints (last 3/4)

**Come monitorare un progetto agile**

Nei progetti Scrum o Kanban, esistono diversi grafici per il controllo e monitoraggio dei progetti:

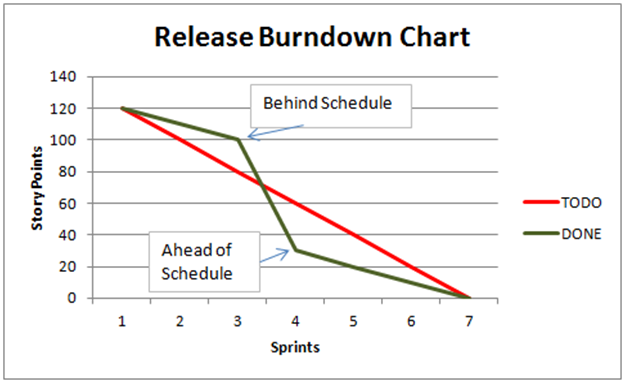
* Burndown chart
* Burnup chart
* Sprint report
* Diagrammi cumulativi

**Burndown chart**

Il Burndown chart, fornisce una chiara indicazione temporale dello sprint, con una barra che identifica il percorso ideale per “bruciare” tutti gli story point di uno sprint.

Viene poi riportato l’andamento attuale del team e dello sprint.

E’ evidente notare un andamento fuori dalle previsioni sia in negativo che in positivo

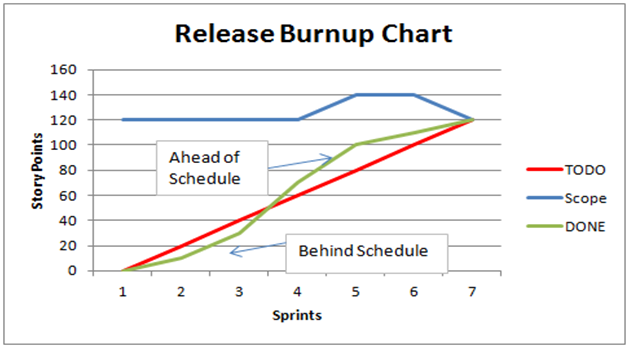


**Burnup chart**

Il Burnup chart, fornisce una chiara indicazione degli story points completati.

Differentemente dal primo, in questo grafico la barra totale degli story point viene mostrata in orizzontale da sinistra a destra, mentre la diagonale indica sempre il percorso ideale.

Mostrando il totale degli story point con la barra orizzontale, è immediatamente visibile in caso ci siano variazioni di scope nello sprint.

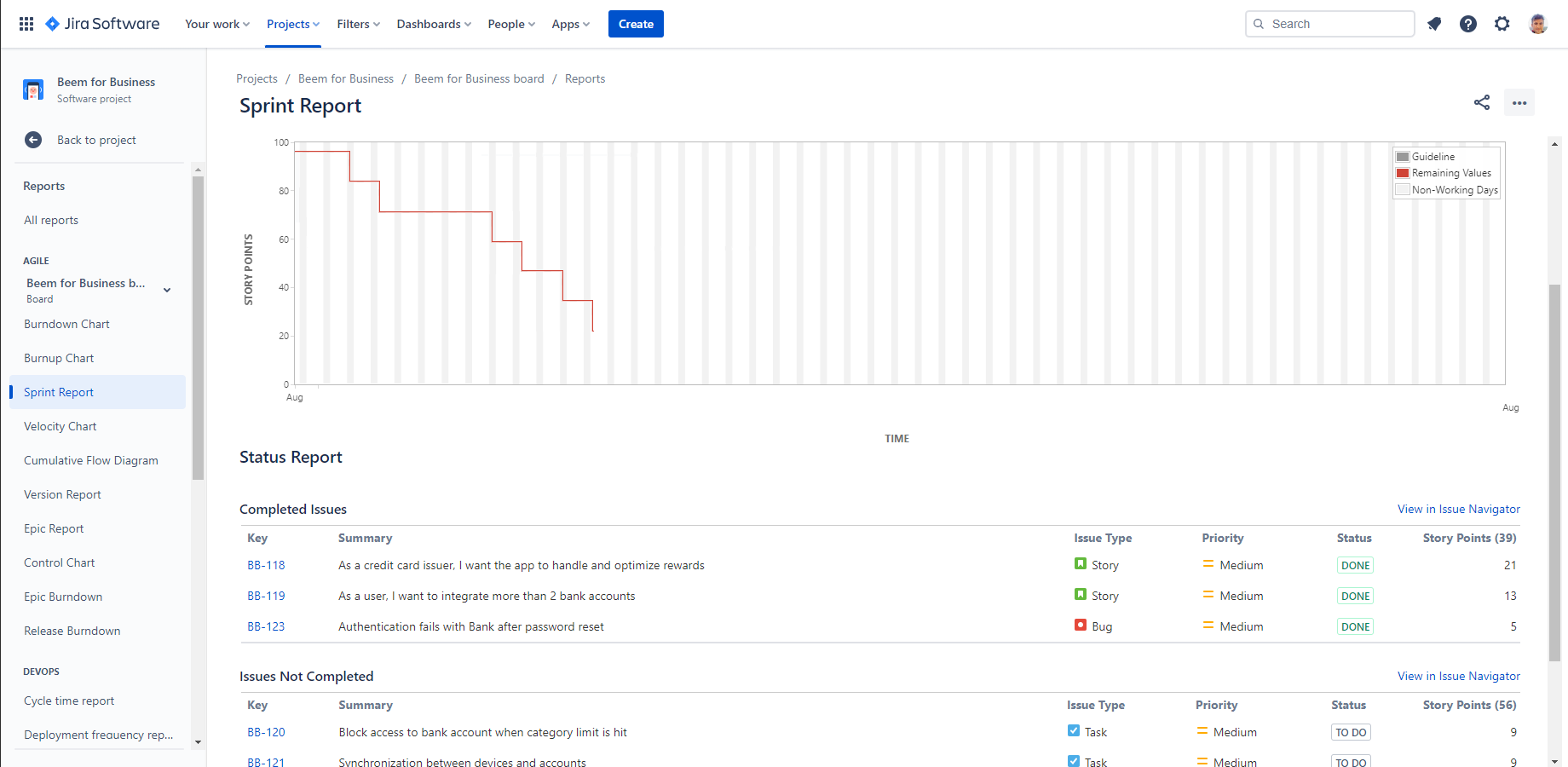


**Sprint report**

Lo Sprint Report mostra l'elenco dei problemi in ogni sprint. È utile per le tue riunioni Sprint Retrospective e anche per i controlli di avanzamento a metà sprint.

Riporta

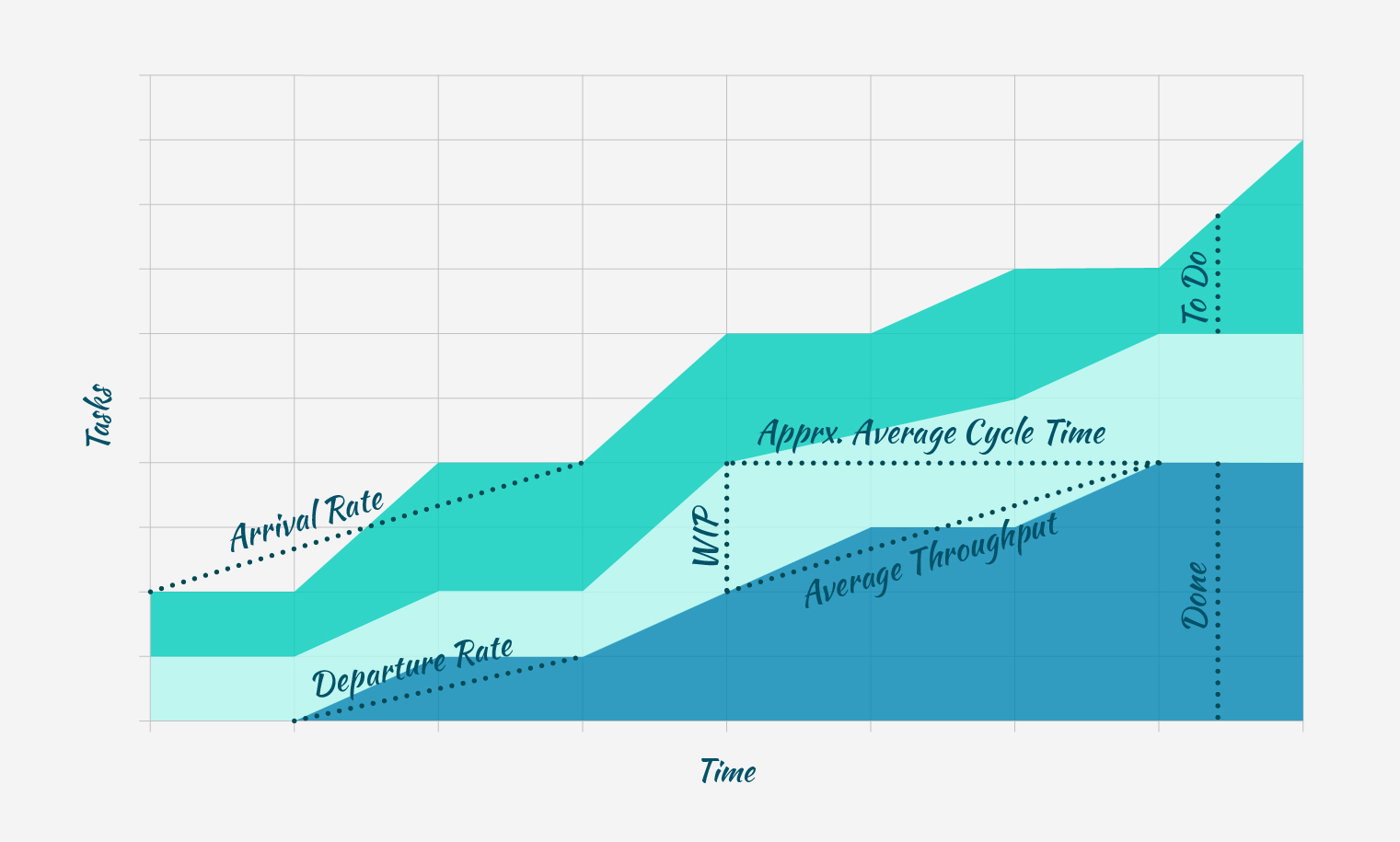
* Grafico Burndown dello sprint
* Resoconto di tutte le attività svolte nello sprint
  + Attività pianificate
  + Attività completate
  + Attività non completate
  + Attività rimosse
  + Attività completate fuori dallo sprint



**Diagrammi cumulativi**

Molto utili per monitorare l’avanzamento di specifiche attività nel tempo:

* Quanti bug creati vs quanti bug risolti
* Quante story aggiunte al backlog vs quante story completate
* Quanti test pianificati vs quanti test automatizzati

****

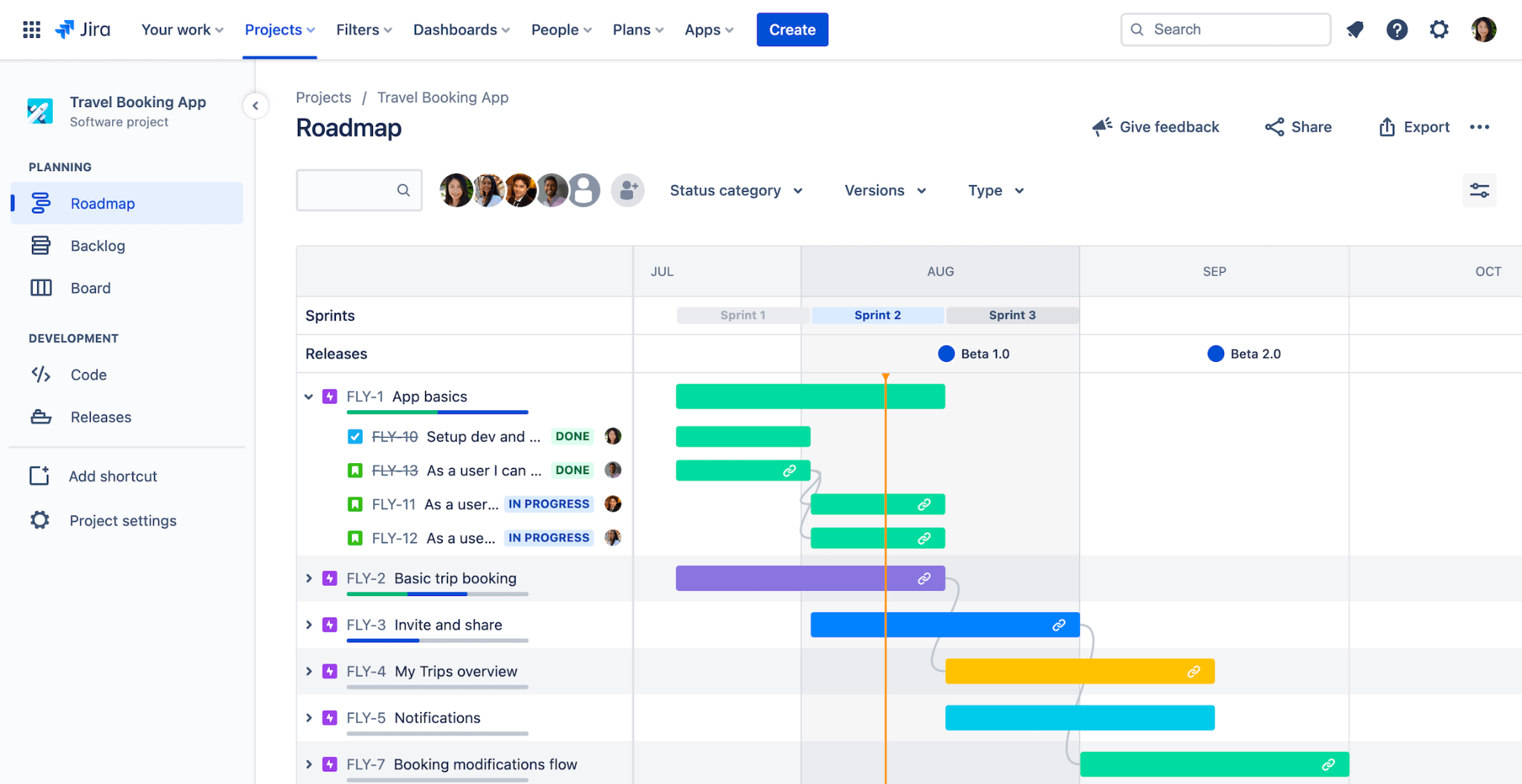
**Diagramma gantt**

Elementi fondamentali di un diagramma di Gantt:

* Linea temporale (suddivisa anche in sprint se possibile)
* Attività nidificate
* Possibili dati su allocazione risorse

Quando è bene usare i diagrammi di Gantt?

* Roadmap
* Programmi
* Gestione di grandi progetti con una durata temporale medio lunga 1+ anni



**SLIDE L5**

**Sommario**

Gli stakeholders

Cosa significa

Tipi di stakeholders

La mappa degli stakeholders

Soft skills

Gestione di un team da remoto

Esercizio

**Stakeholders**

Ciascuno dei soggetti direttamente o indirettamente coinvolti in un progetto o nell'attività di un'azienda.

Stakeholder

* Interni
* Esterni

**Stakeholder esterni**

* Clienti
* Popolazione
* Creditori
* Fornitori
* Governo

**Stakeholder interni**

* Proprietari
* Investitori
* Board aziendale
* Manager
* Dipendenti

**Mappa degli stakeholder**

La mappa degli stakeholder è un grafico efficace per identificare gli stakeholder chiave relativi al progetto.

Aiuta a conoscere e identificare tutte le persone interessate al risultato del progetto.

La mappa degli stakeholder è divisa in 4 quadranti suddivisi in base ai valori: potere e interesse.

I 4 quadranti sono:

* Da soddisfare - Alto potere / Basso interesse
* Da gestire attentamente - Alto potere / Alto interesse
* Da tenere informati - Basso potere / Alto interesse
* Da monitorare - Basso potere / Basso interesse

**Esempi di stakeholders**

Da soddisfare - Alto potere / Basso interesse

* Direttori aziendali
* Dipartimento Sales

Da gestire attentamente - Alto potere / Alto interesse

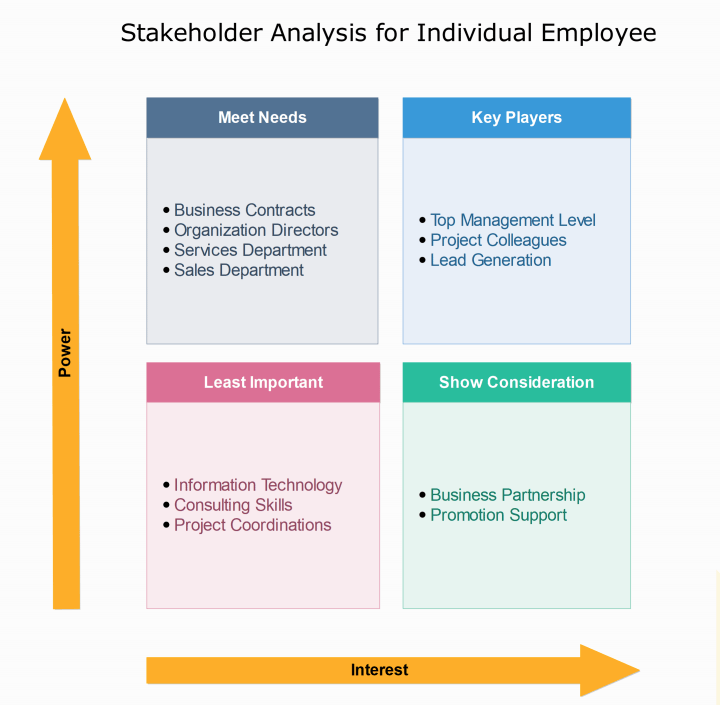
* Board aziendale
* Partner di progetto

Da tenere informati - Basso potere / Alto interesse

* Partner commerciali

Da monitorare - Basso potere / Basso interesse

* Dipartimento IT
* Consulenti
* Coordinatori di progetto



**Soft skills**

Il concetto di Soft Skill fa riferimento a quelle competenze legate all'intelligenza emotiva e alle abilità naturali che ciascuno di noi possiede.

Le Soft Skills non riguardano delle competenze tecniche, ma piuttosto sono legate a come interagisci con i colleghi, risolvi i problemi e a come gestisci il tuo lavoro.

In altre parole, si tratta di tutte quelle competenze trasversali che nella vita professionale sono essenziali.

**Elenco soft skills per un project manager**

* Leadership - capacità di guidare
* Motivation - capacità di motivare
* Communication - capacità di comunicare in modo efficace
* Conflict Management - capacità di comprendere, gestire e risolvere i conflitti
* Trust-Building - capacità di instaurare fiducia
* Decision-Making - capacità di prendere decisioni
* Organization - capacità di organizzare
* Time Management - capacità di gestione del tempo

**Gestione di un team da remoto**

Elementi per gestire un team da remoto:

* Obiettivi chiari e condivisi -  Sprint - Roadmap - OKRs (Objective key results)
* Strumenti di controllo e monitoraggio chiari e semplici - Sprint board - Report
* Momenti di lavoro alternati a momenti di team building - Meeting informali - Attività di gruppo
* Retrospective condotte regolarmente e seriamente - Ogni sprint - Trimestre - Anno
* Fiducia nel team e fra i vari componenti - Costruire un ambiente basato sulla fiducia fra le persone
* Dare feedback e riconoscere i meriti - Premiare in gruppo e riprendere in privato

**Strumenti di gestione per un team da remoto**

* Jira software
* Confluence
* Slack
* Google Drive
* Parabol