

# Introduzione a Scrum

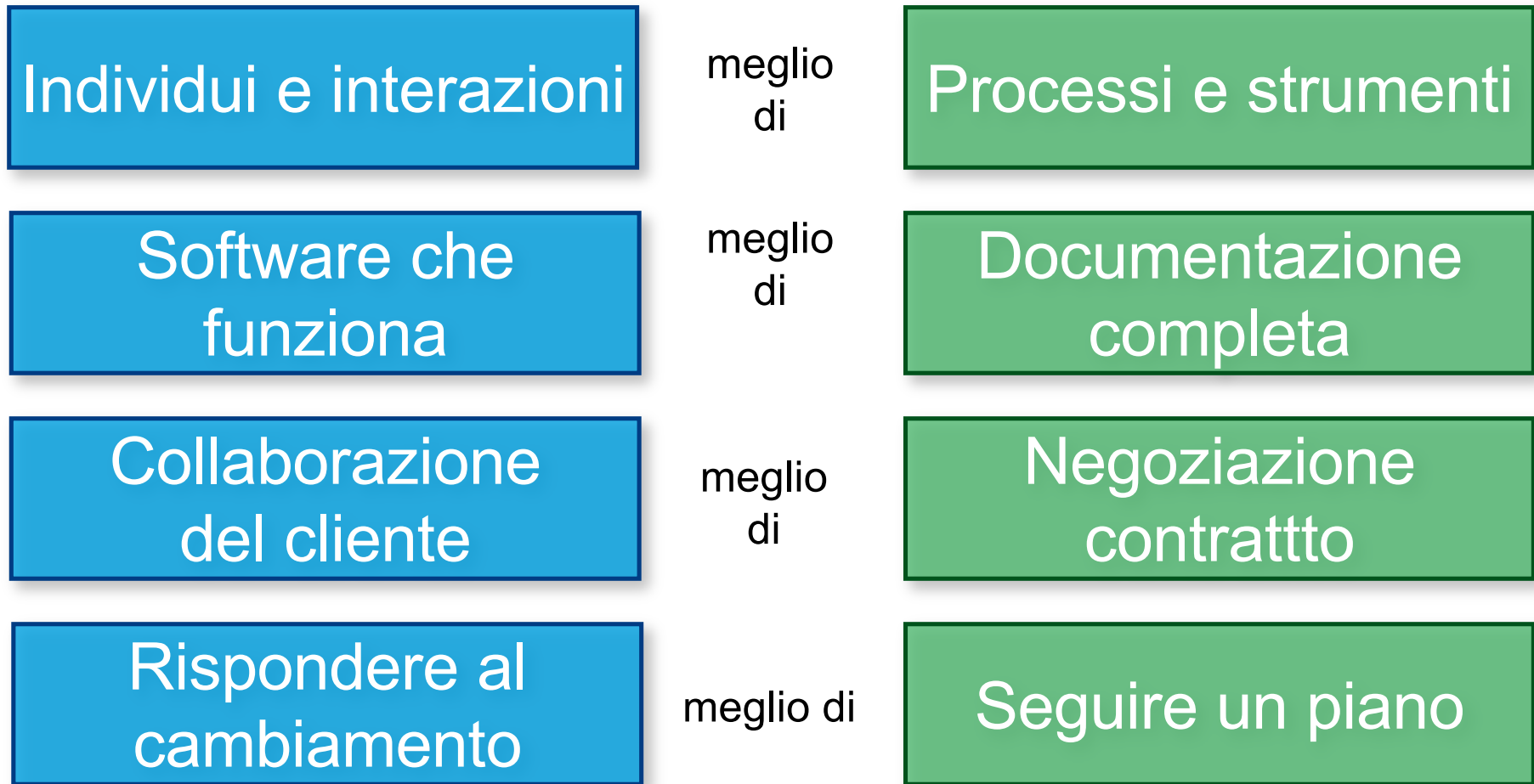


*Corso di Ingegneria del Software*  
*CdL Informatica*  
*Università di Bologna*

# Obiettivi della lezione

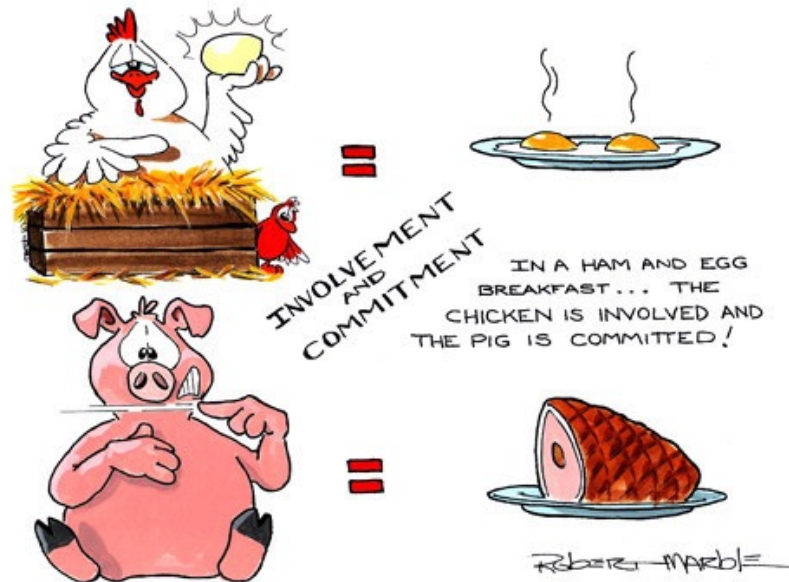
- Presentazione del metodo agile Scrum
- Ruoli
- Rituali
- Artefatti
- Varianti di Scrum

# Manifesto: valori “agili”



Fonte: [www.agilemanifesto.org](http://www.agilemanifesto.org)

# Galline e maiali





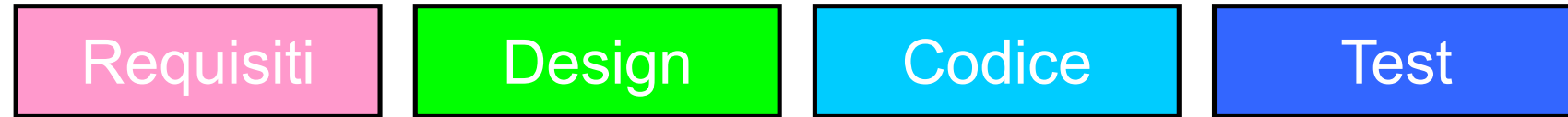
# Staffetta o pacchetto di mischia?

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

Hiroataka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”, *Harvard Business Review*, January 1986.



# Dalla staffetta al pacchetto di mischia



Non una cosa alla volta

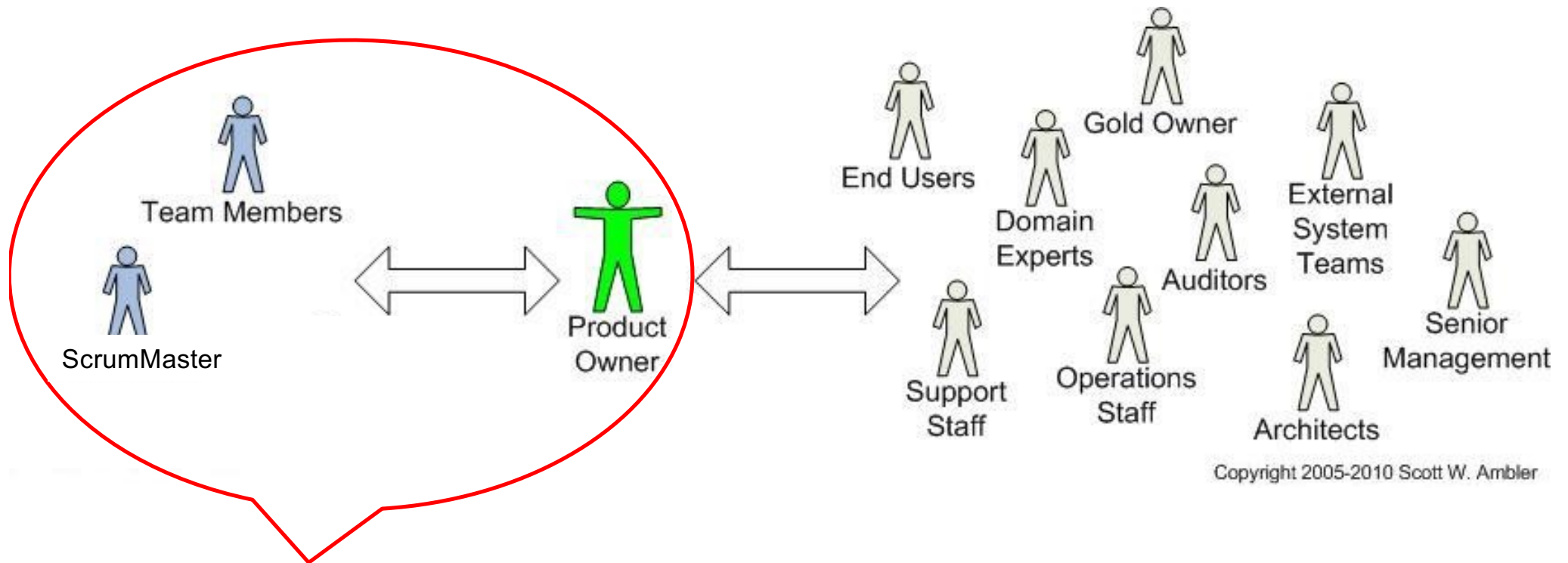
il team scrum fa un po' di tutto in ogni sprint

Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

## Scrum in 100 parole

- Scrum è un modello di processo per produrre software ottenendo il massimo valore utile nel minor tempo
- Il cliente definisce le funzioni da realizzare e loro priorità
- Il team di sviluppo decide quotidianamente il modo migliore di produrre le funzioni di più alta priorità.
- Alla fine di ogni sprint nasce una nuova versione che viene esaminata dal cliente per decidere se continuare lo sviluppo con un altro sprint e/o produrre un rilascio

# Scrum: ruoli pig e ruoli chicken



Ruoli "*pig*"

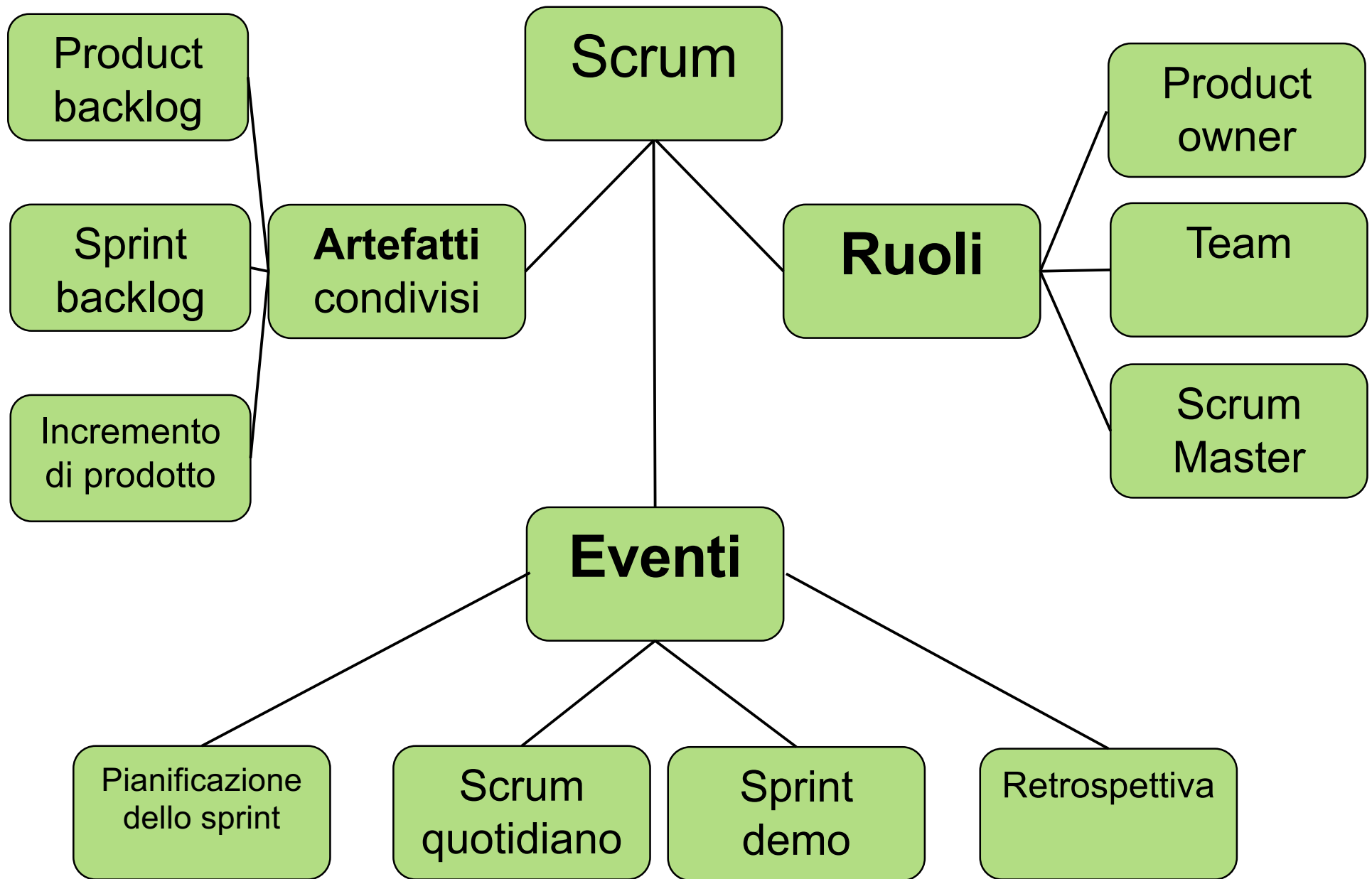
Ruoli "*chicken*"

# Valore agile: trasparenza

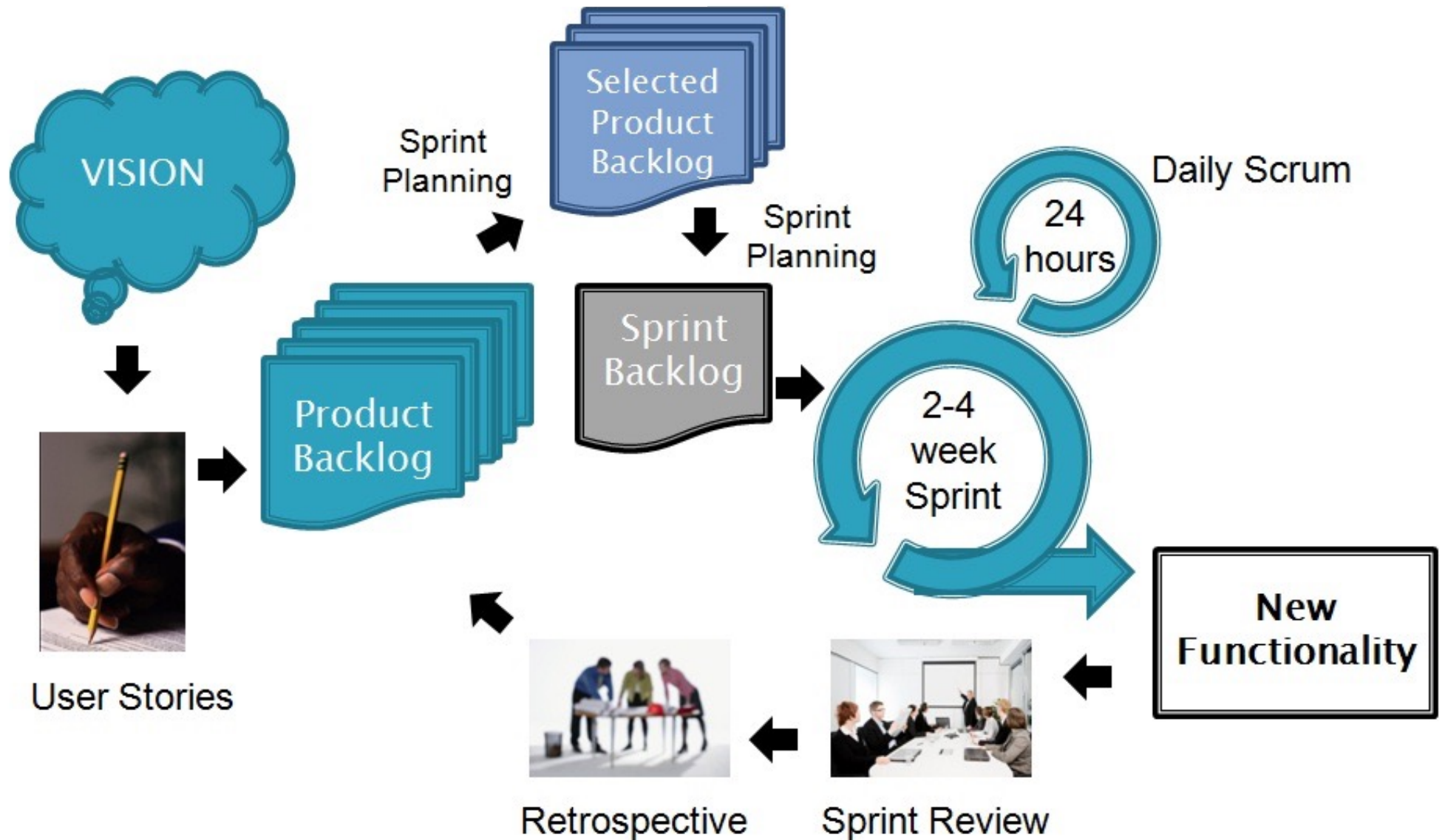
- Gli aspetti significativi del processo devono essere **visibili** ai responsabili del risultato finale (i “*pigs*”: il Product Owner ed il Team)
- La trasparenza richiede la condivisione e la comune comprensione di ciò che viene visto

Esempi:

- Tutti i partecipanti debbono condividere un linguaggio comune di riferimento al processo di sviluppo
- La definizione di ciò che è “Fatto” deve essere condivisa sia da chi esegue sia da chi deve accettare un task



# Scrum: gli eventi e gli artefatti condivisi



# Scrum

- Ideato da Schwaber e Sutherland [OOPLSA 1995]
- Una strategia di sviluppo di prodotti per cui un team lavora unito per raggiungere un obiettivo comune
- Iterativo e incrementale, basato su controllo empirico del processo



# Punti chiave di Scrum

- Team di sviluppatori auto organizzante
- Il Product Owner si assicura che venga costruito il prodotto “giusto”
- Lo Scrum Master si assicura che non ci siano impedimenti al processo di sviluppo
- Sviluppo guidato da storie e test
- Il prodotto cresce in “sprint” di durata fissa
- Per ogni sprint, ogni membro del team sceglie i requisiti da realizzare da uno “sprint backlog”
- Eventi (meeting): sprint planning, scrum quotidiani, demo reviews e “retrospettive”

# User story: forma basica

Come **soggetto**

Voglio **una funzione**

Allo scopo di **valore per il soggetto**

**Esempio**

Come **giocatore**

Voglio **poter iniziare una partita a scacchi**

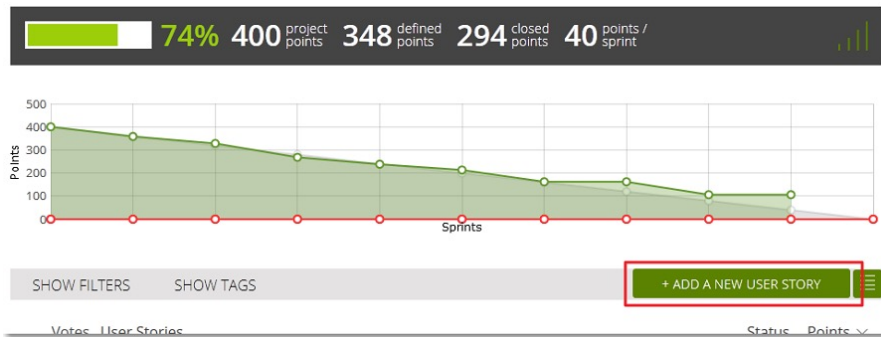
Per **giocare in rete contro un avversario qualsiasi**

Il backlog di prodotto è una lista di storie ordinate dal PO e stimate dal team

<b>ToDo List</b>		
<b>Story</b>	<b>Estimation</b>	<b>Priority</b>
As a user I want to be able to reset my password	1	1
As a user I want to edit items	3	2
As a user I want to export data	2	3
As an administrator I want to define KPI's for my sales team	4	4
As a user I want to view my data on mobile	5	5
As an administrator I want to send alerts when new leads come in	2	6
As a user I want to create a report of my data	5	7
As a user I want to update my reminder settings when a date is added	3	8
As a user I want filtering enhancements	4	9
As an administrator I want to configure views of data	5	10
<b>Total</b>	<b>34</b>	

# user story su Taiga

<https://taiga.pm/creating-a-user-story-from-the-backlog/>



## New user story

Subject

? TW ? Design ? Front ? Back

? Product Owner

? total points

New

I'm it! Tag me...

Please add descriptive text to help others better understand this US

0 attachments +

Drop attachments here!

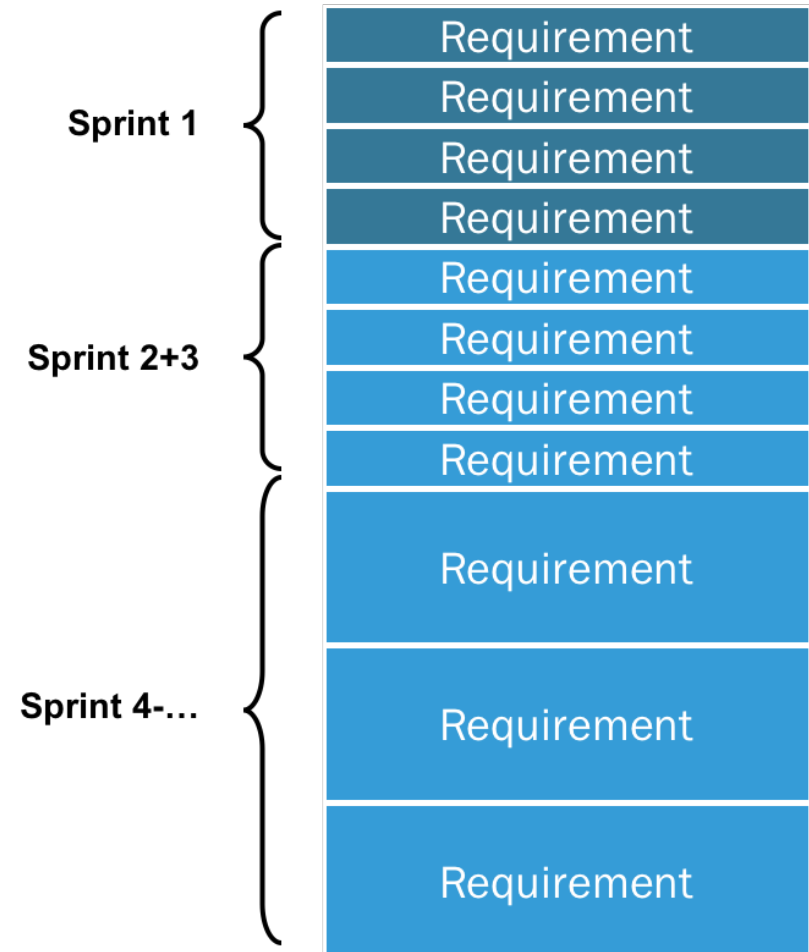
TEAM REQUIREMENT CLIENT REQUIREMENT BLOCKED

CREATE

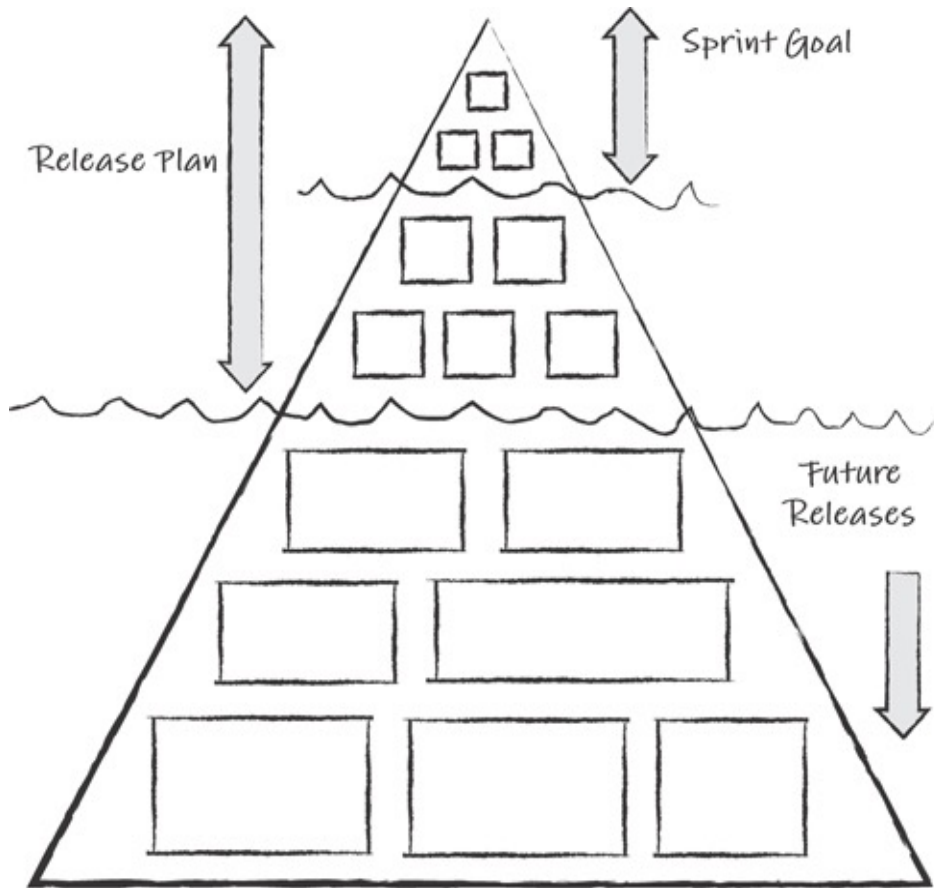
<https://taiga.pm/the-taiga-panel/>

# Il backlog di prodotto è ordinato

- L'ordinamento (priorità) delle user story è compito del PO
- La suddivisione di quali storie verranno realizzate nel prossimo sprint è compito del team, sulla base delle stime di sviluppo
- (le user story sono requisiti)

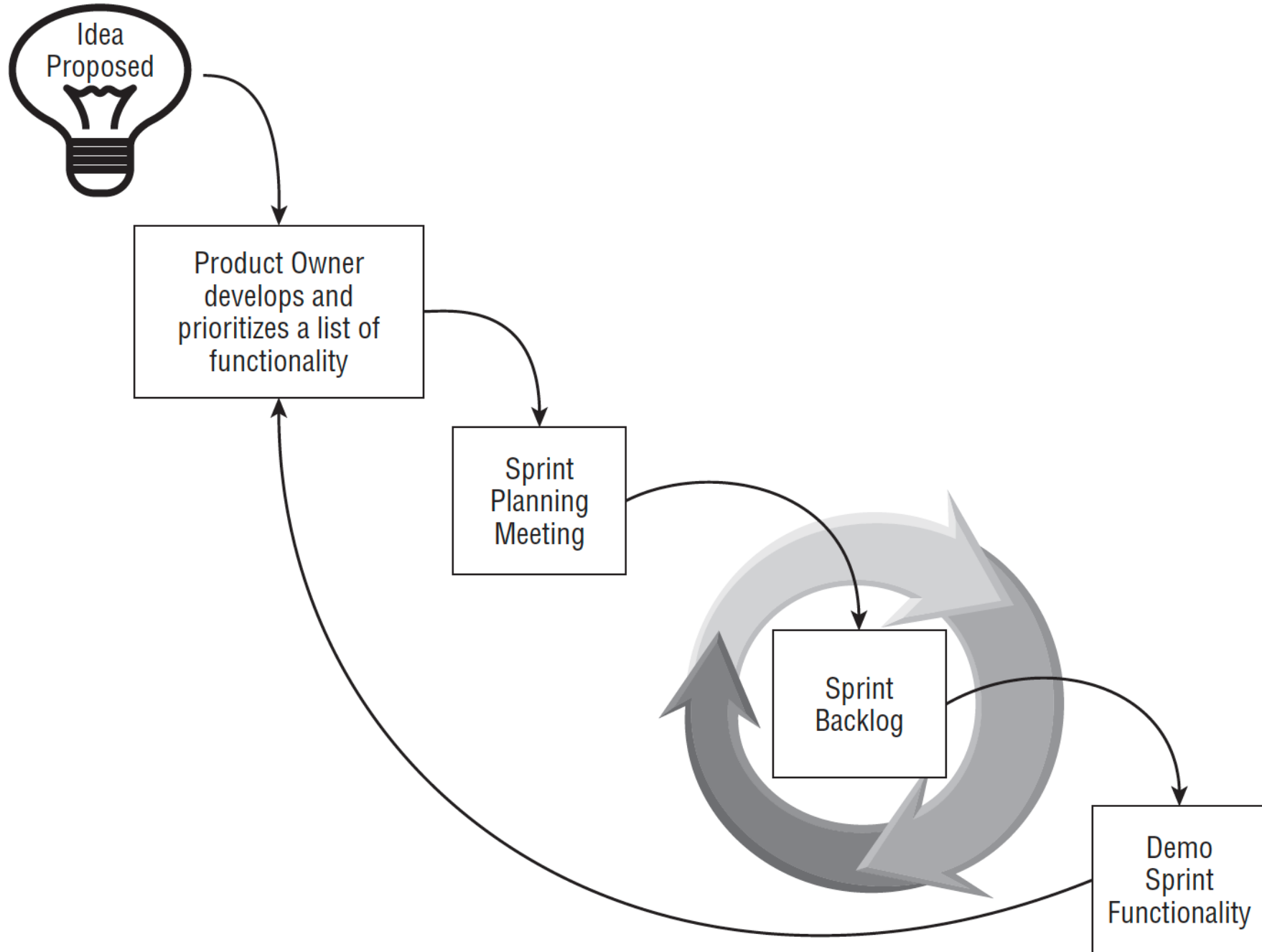


# Il product backlog è un iceberg

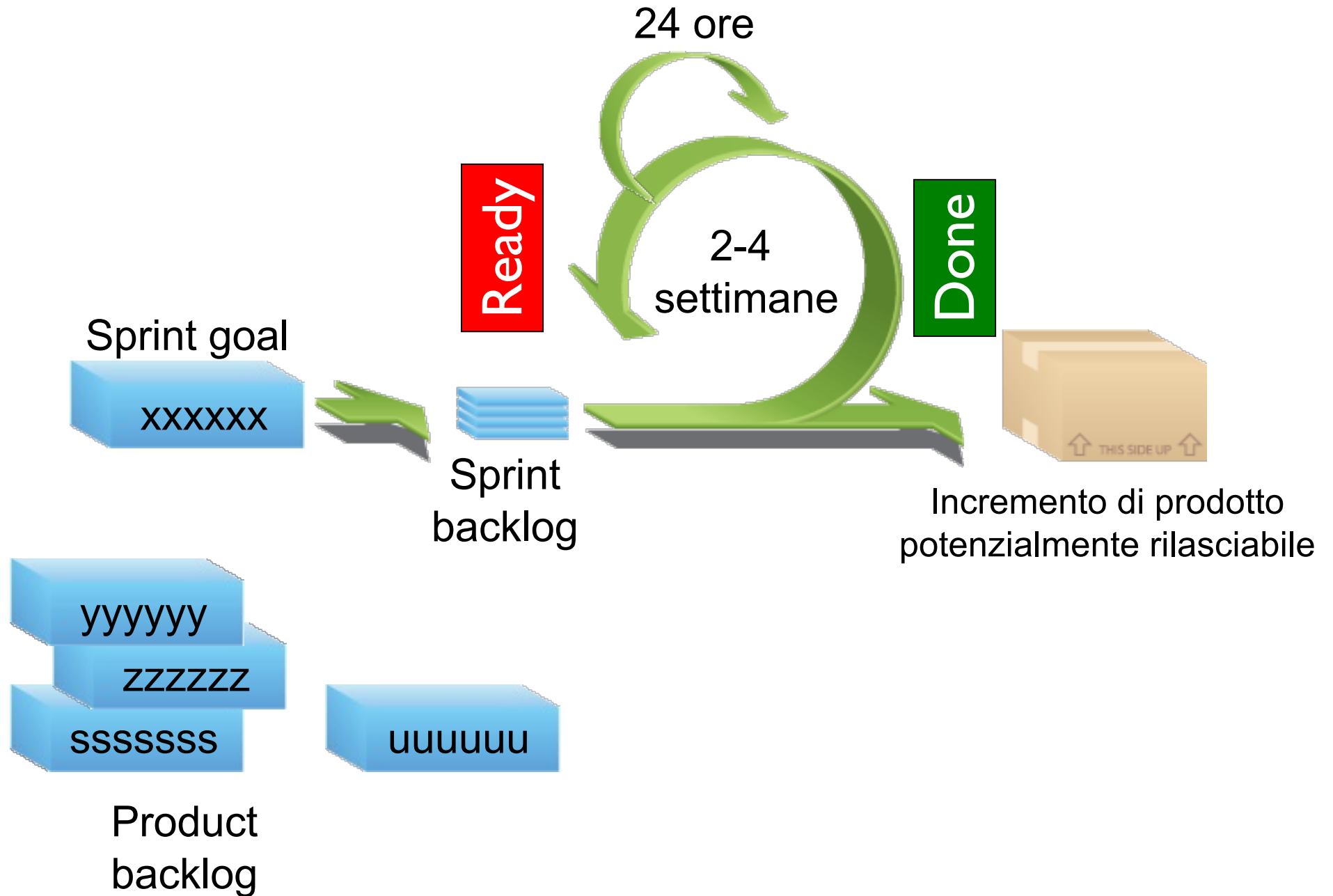


- La cima dell'iceberg sono le storie pronte per essere realizzate
- Le storie sotto la cima sono previste nel piano dei rilasci, anche se sono ancora poco chiare
- Sotto la superficie ci sono i rilasci futuri, ancora non concordati o nemmeno immaginati

# Il ciclo di base in Scrum



# Un processo Scrum si compone di **sprint**

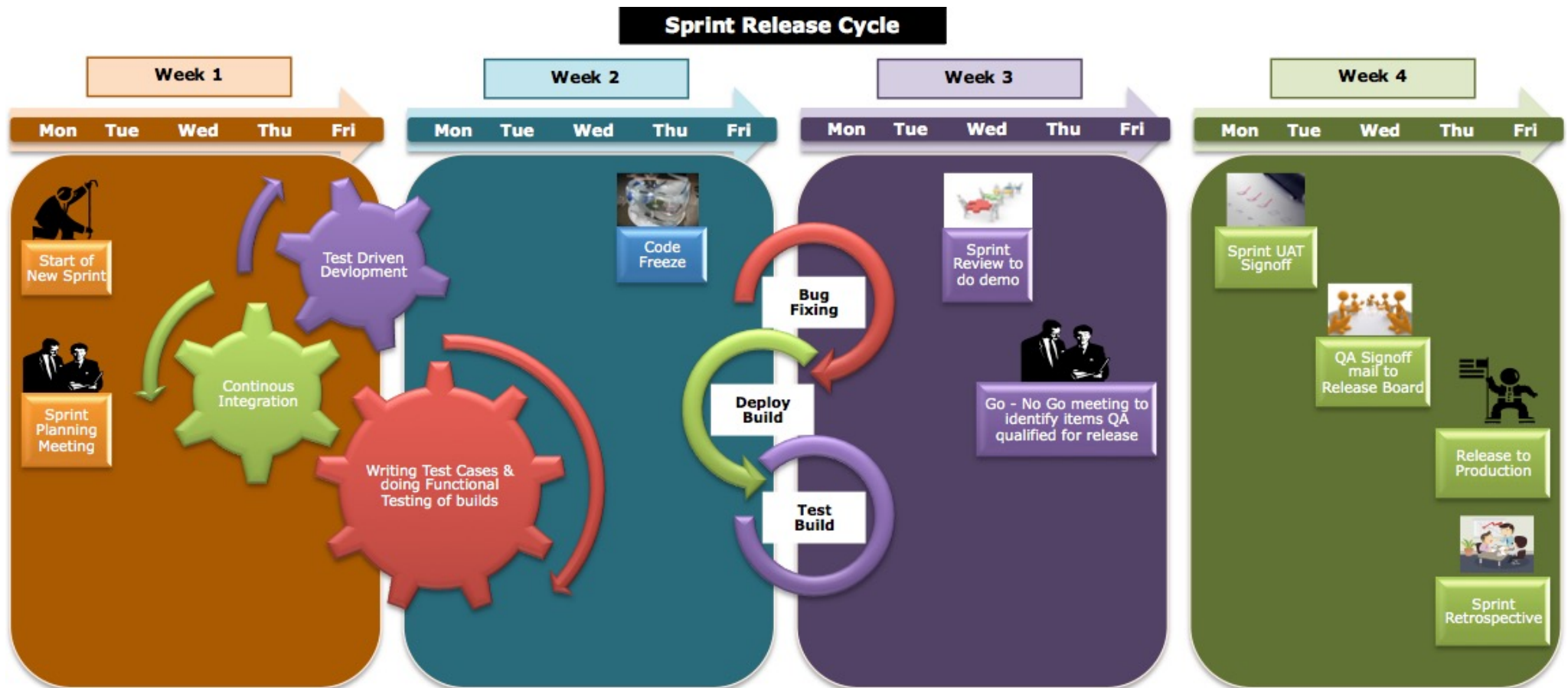




# L'iterazione principale: lo sprint

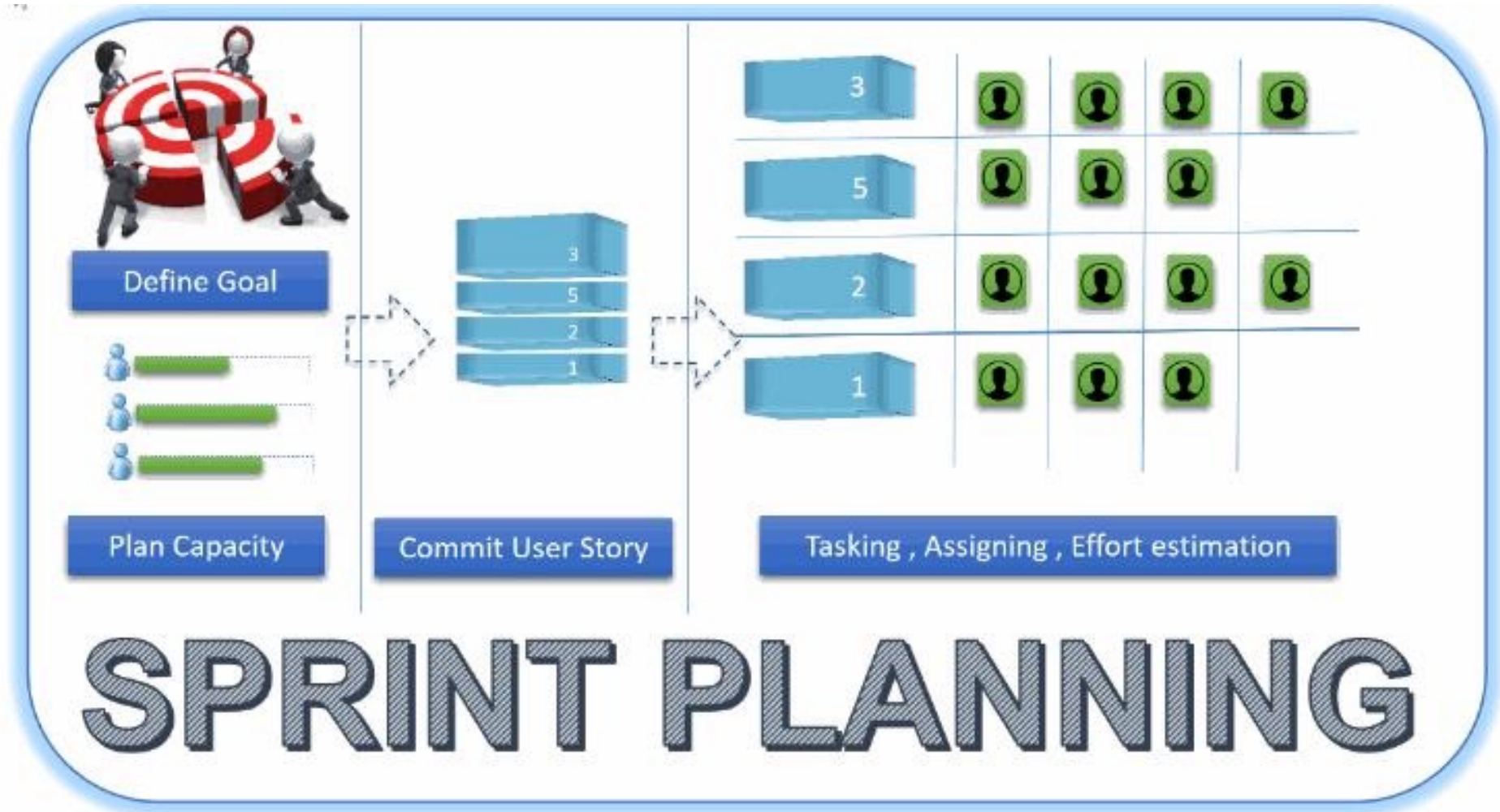
- Un processo Scrum è una serie di sprint (iterazioni)
- Durata di ogni sprint da 2 a 4 settimane (a scelta del team+PO)
- Durata di ogni sprint: costante (migliora il ritmo del team)
- Ogni sprint include: design, codifica e test
- Ogni sprint estrae funzioni “Ready” dal product backlog e aggiunge codice “Done” al prodotto, da mostrare al cliente

# Ciclo di sprint (ogni sprint un rilascio)



Fonte: [www.scrumalliance.org](http://www.scrumalliance.org)

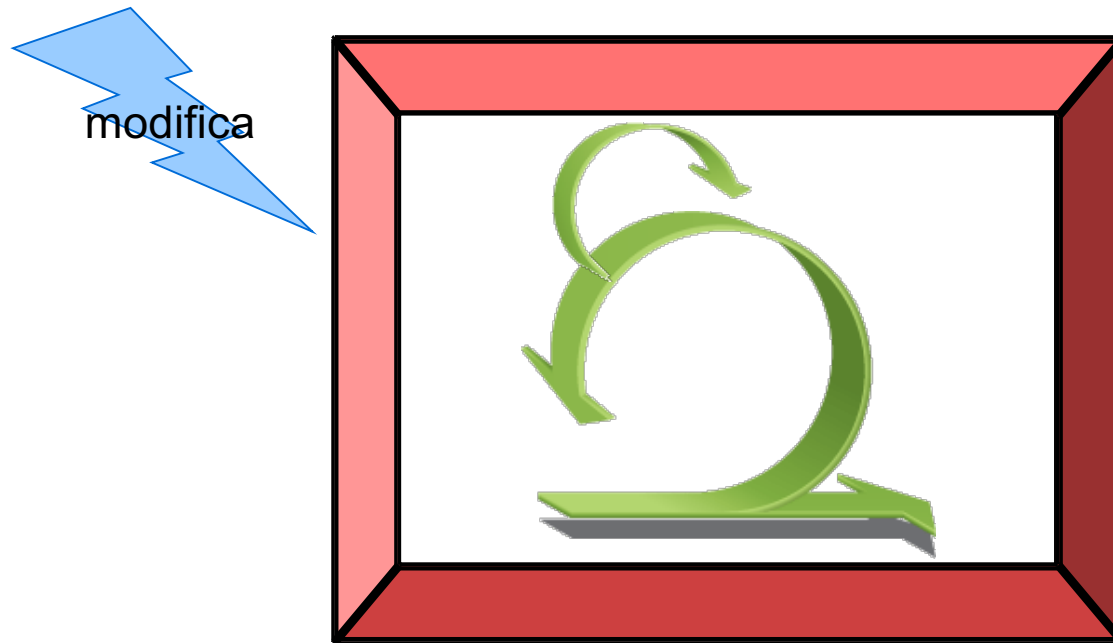
# La pianificazione dello sprint



# Pianificare lo sprint

- Ogni sprint inizia con una pianificazione ( a cura di PO e Team) che dura almeno un'ora per ciascuna settimana di sprint
- Tale pianificazione ha per risultato la scelta di alcune user story che formano il backlog di sprint
- Il Team stima quanto lavoro riuscirà a fare durante lo sprint, tenendo presente la sua «velocità» di sviluppo
- Il Team decide quale parte del backlog di prodotto viene estratta per diventare il backlog di sprint: le user story di maggior priorità vengono scelte assieme al PO
- Viene concordata da PO e Team la «Definition of Done» (DoD) per tutti gli elementi del backlog di sprint: il PO accetta come completate solo le US che soddisfano la DoD

# Niente cambiamenti al team durante lo sprint



Pianificare le durate degli sprint in modo da garantire che il team non cambi

# Scrum: ruoli eventi artefatti

## Ruoli

- Product owner
- ScrumMaster
- Team

## Eventi

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

# Scrum: i ruoli

## Ruoli

- Product owner
- ScrumMaster
- Team

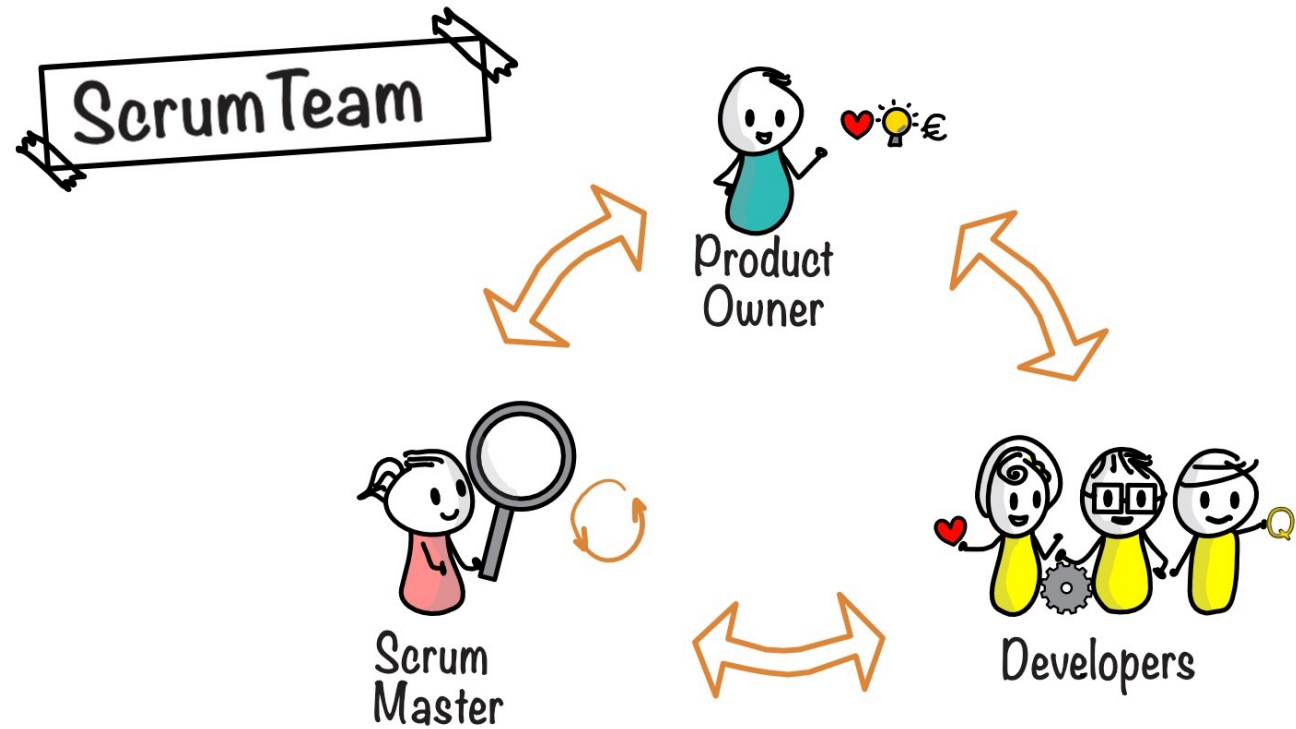
## Rituali

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

# I ruoli principali in Scrum

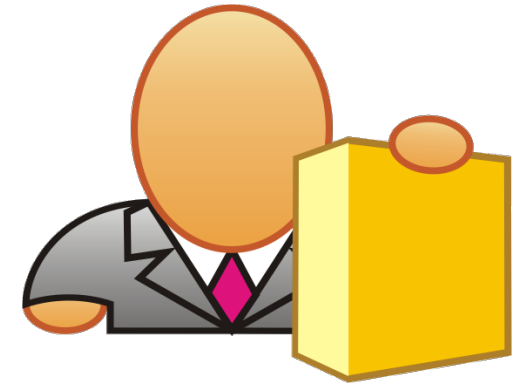




# I ruoli in Scrum

- **Product Owner (PO):** rappresenta gli stakeholder (la voce del cliente), scrive il *product backlog* in forma di *user stories*; è responsabile della definizione di “Fatto”.
- **Team di sviluppo:** 3-9 membri con diverse abilità, collettivamente responsabili della consegna di un PSI (Potentially Shippable Increment)
- **Scrum Master (SM):** facilita la corretta esecuzione del processo, elimina gli ostacoli
  - Product Owner e Scrum Master debbono essere persone diverse; possono collaborare allo sviluppo
  - SM non ha responsabilità di gestione del personale o del progetto in senso “tradizionale” – es. non dà premi o sanzioni

# Product Owner (PO)



- Responsabile del valore del prodotto (Return On Investment - ROI)
- Definisce le feature (funzioni) del prodotto
- Decide i rilasci: date e contenuti
- Mette in priorità le feature rispetto al valore di mercato
- Per ogni iterazione rivede lista delle feature e loro priorità ove necessario
- Accetta o rifiuta i risultati dello sviluppo mediante la definizione di “Fatto”

# Il Product Owner in Essence

«Possiede» il prodotto

Gestisce la visione del prodotto e la sua evoluzione

Mostra il prodotto agli stakeholder

Acquisisce l'accettazione del prodotto

## Product Ownership Essentials

Own, evolve and communicate the vision, and guide the evolution of the product to achieve the vision.

  
Product  
Ownership


  
Build  
Stakeholder  
Network

  
Stakeholder  
Network

  
Evolve the  
Product Vision

  
Product Vision

  
Demonstrate  
the Product

  
Achieve  
Acceptance

  
Resources



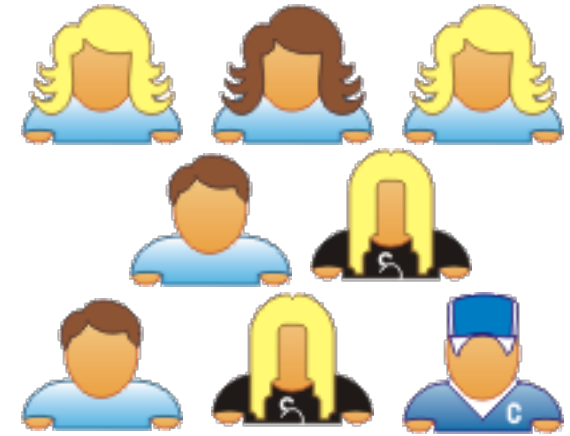
# Il team di sviluppo



- Auto-organizzante
- Modifiche al team solo in sprint diversi: è stabile per tutto lo sprint
- Tutti presenti nello stesso spazio di lavoro
- Include uno Scrum Master come coach e mentore

# I membri del team

- 5-9 persone
- Varie specialità:
  - Programmatori, testatori, user experience designers, database administrator, ecc.
- Impegno full-time
  - Con eccezioni (e.g., database administrator, costoso)



# Caratteristiche del team Scrum

Il team include sviluppatori capaci di produrre un incremento del prodotto “Fatto” alla fine di ogni sprint.

Il team si autogestisce ed ha le seguenti caratteristiche:

- I membri si auto-organizzano; nessuno, nemmeno lo Scrum Master, può dire al Team come trasformare il Product Backlog in incrementi di funzionalità potenzialmente rilasciabili;
- I team sono cross-funzionali, cioè includono tutte le abilità necessarie a creare un incremento di prodotto;
- I membri del Team sono tutti Sviluppatori alla pari, senza nessuna eccezione;
- Il Team è unitario e non include sottogruppi, senza nessuna eccezione per attività o domini particolari;
- I membri del Team possono avere qualche specializzazione personale, ma l'intero Team resta responsabile dello sviluppo.

# Il Team in Essence

Il team si autoorganizza

Include un facilitatore  
(detto Scrum Master)

Sceglie una modalità di  
lavoro

Collabora e condivide la  
conoscenza

Tiene un passo sostenibile

## Agile Teaming Essentials

A self-organizing team maximizes its performance by using a highly-collaborative teaming approach.



Self-Organizing Team



Agile Facilitator



Team Kick-Start



Team Membership



Team Way of Working



Sustainable Pace



Collaborate, Communicate, Coordinate



Share Knowhow



Collaboration Space



Resources



# ScrumMaster

- Serve il team, non lo dirige
- Responsabile dei valori e pratiche agili
- È il “Process Owner”: coordina i meeting e rimuove gli impedimenti al processo Scrum
- Assicura il benessere del team
- Supporta la cooperazione di ruoli e funzioni
- Protegge il team da interferenze esterne
- Non è detto che sia un programmatore





# Mappa concettuale delle responsabilità di SM



# I ruoli Scrum: le responsabilità



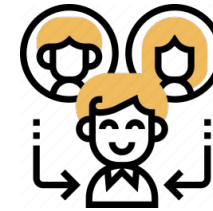
Costruire  
il prodotto  
giusto

Product owner



Costruire  
il prodotto  
bene

Sviluppatori



Evitare gli  
inciampi e  
assicurare  
la velocità

Scrum Master

# Scrum: gli eventi

## Ruoli

- Product owner
- ScrumMaster
- Team

## Eventi

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

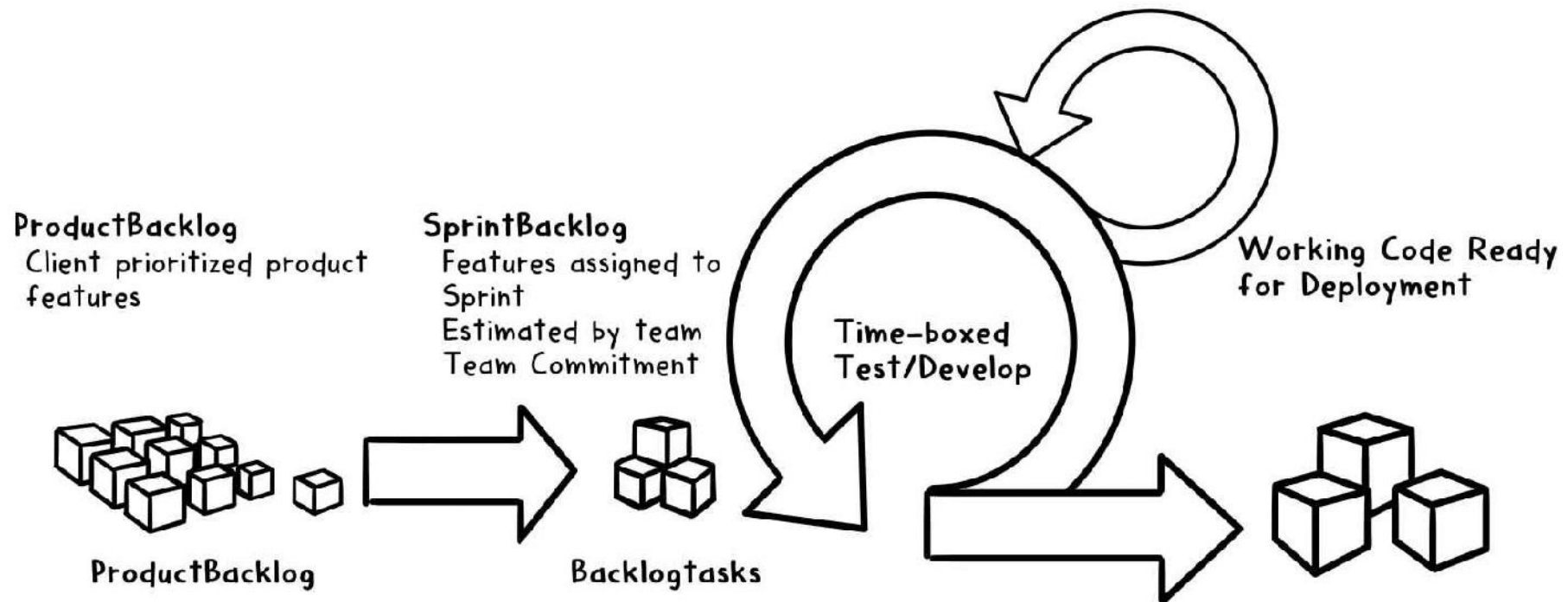
## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

# Gli eventi Scrum

- **Sprint planning:** PO e Team decidono cosa fare (sprint backlog) e come aggiornare il product backlog
  - Durata: circa un'ora per ogni settimana dello sprint
- **Daily scrum o stand-up:** ogni sviluppatore dice cosa ha fatto, cosa pianifica per oggi, che impedimenti ha trovato
  - Durata: Max 15 minuti, in piedi
- **Sprint Review:** riguarda il prodotto: cosa è stato o non è stato completato in questo sprint, demo
  - Durata: 4 ore al massimo
- **Sprint Retrospective:** riguarda il processo: cosa è andato bene e quali impedimenti sono stati trovati
  - Durata: 3 ore al massimo

# Gli eventi Scrum



## Sprint Planning Meeting

- Review Product Backlog
- Estimate Sprint Backlog
- Commit

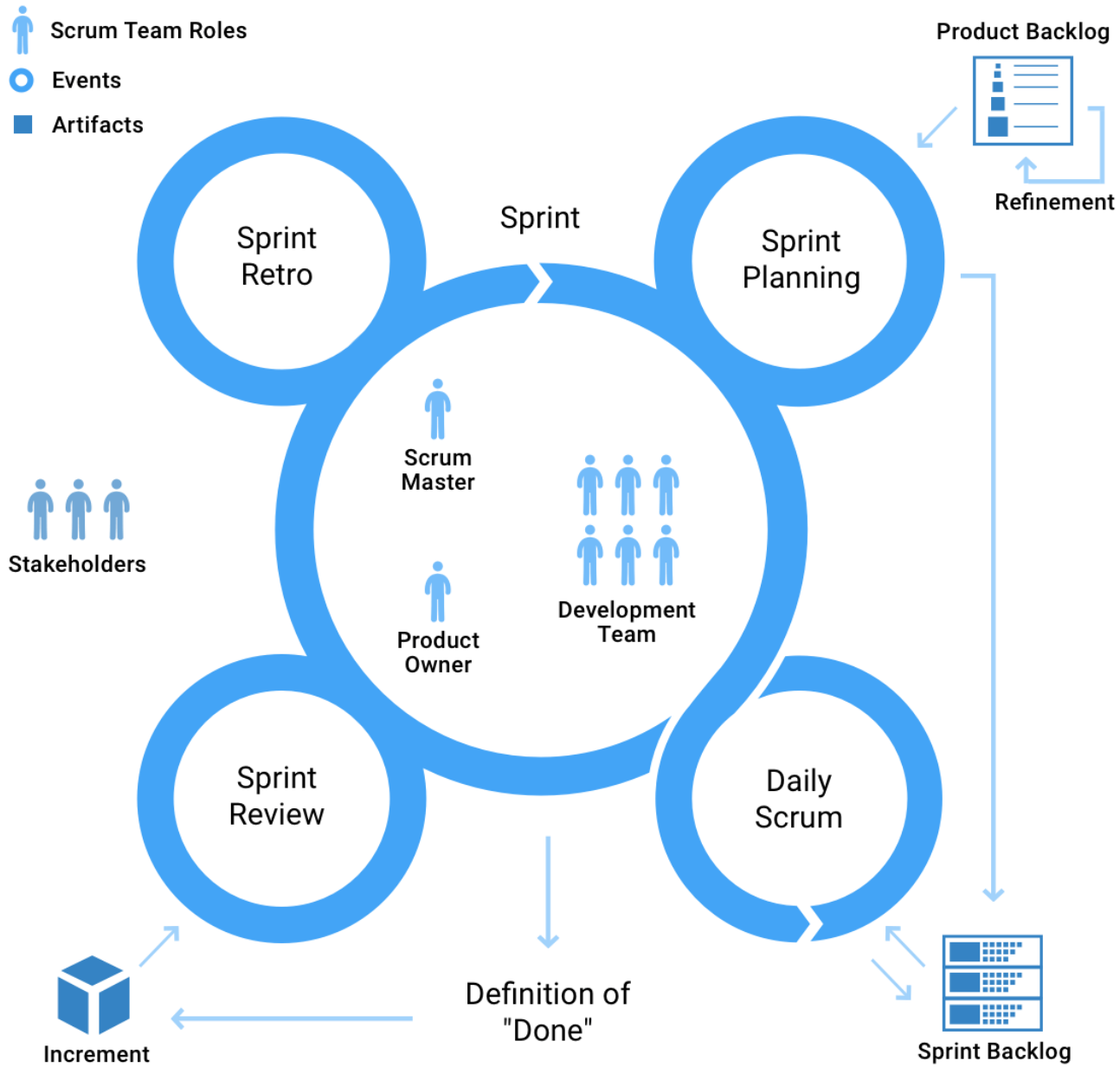
## Daily Scrum Meetings

- Done since last meeting
- Plan for today
- Roadblocks/Accelerators?

## Sprint Review Meeting

- Demo features to all
- Retrospective on the Sprint
- Adjustments

# Scrum



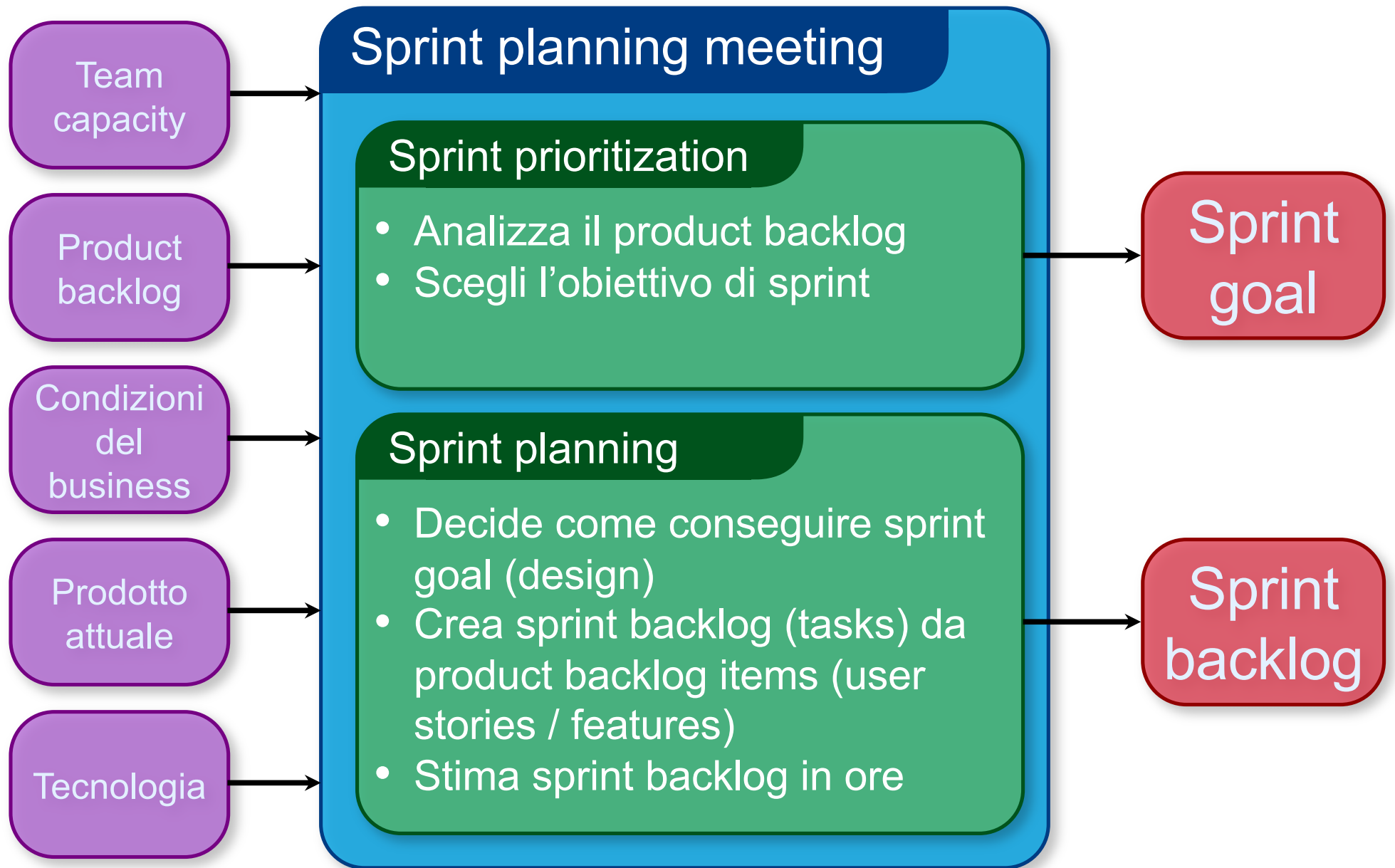
# Struttura quotidiana di uno sprint su base settimanale

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
SPRINT PLANNING 2 HRS.	STAND-UP 15 min.	STAND-UP 15 min.	STAND-UP 15 min.	STAND-UP 15 min.
		STORY TIME 1 HR.		SPRINT Review 1/2 HR.
				RETROSPECTIVE 90 minutes

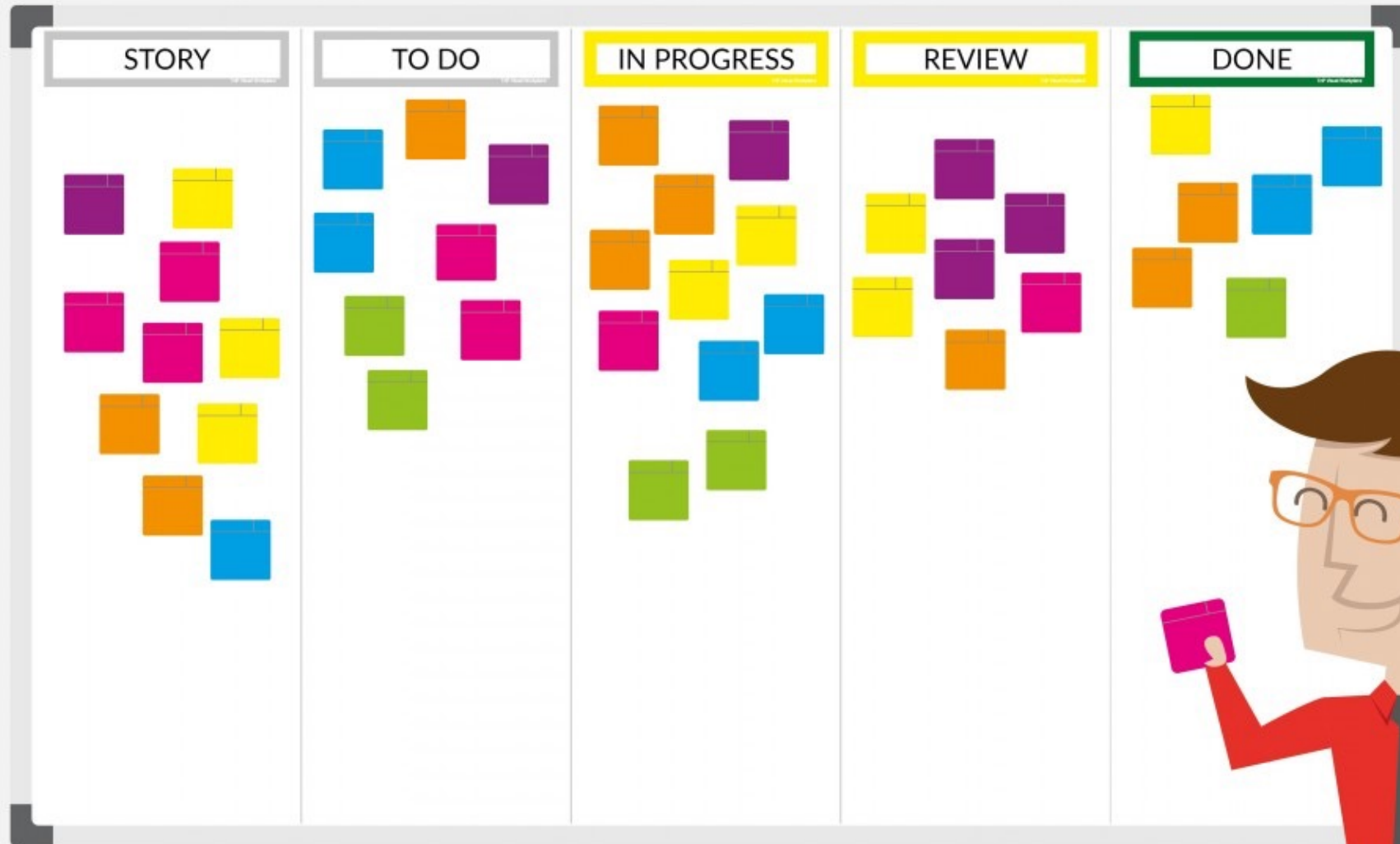
# Il meeting di pianificazione dello sprint







# The scrum board

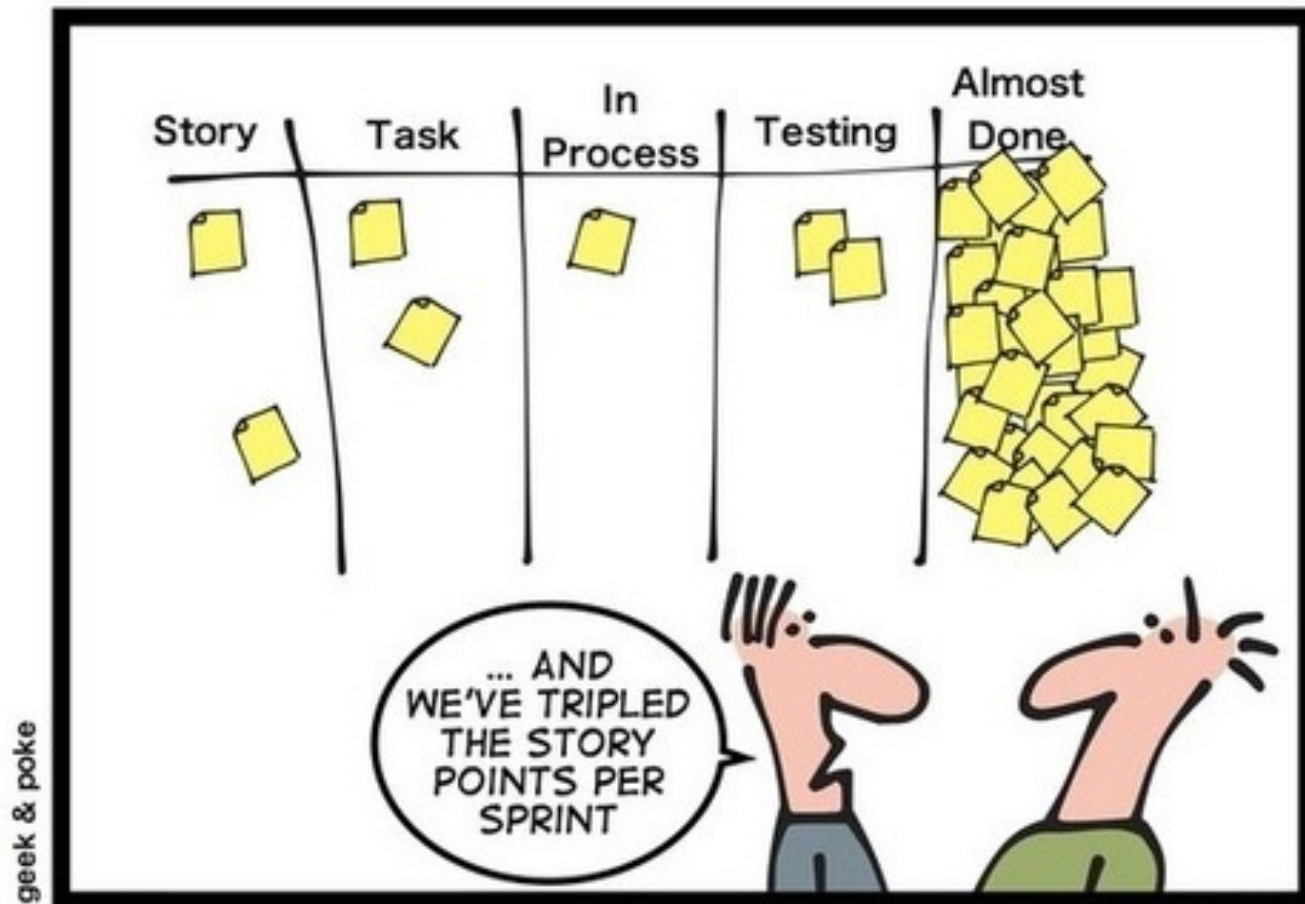


# La definizione di Pronto (definition of Ready)

Una storia è Ready quando:

- Tutti i membri del team la capiscono
- È dettagliata abbastanza da poter essere poi testata
- Il PO deve comprenderne il valore per poterla ordinare in priorità con le altre
- Il Team dev'essere in grado di farne una stima in modo da completarla entro uno sprint
- La storia è indipendente dalle altre: il Team può lavorarla avendone il pieno controllo

# La definizione di Fatto (Definition of Done)



DOAD

# Esempio di definizione di “Fatto”

La lavorazione di una funzione che realizza una user story non è finita a meno che non soddisfi il PO: la definizione di “Fatto” è concordata dal Team col PO all’inizio del processo

Una possibile definizione di Fatto:

- i. Codifica della funzionalità richiesta: completata
- ii. Test di unità: scritti ed effettuati
- iii. Test di integrazione: superato
- iv. Test prestazionale: superato
- v. Documentazione (minimale): scritta
- vi. Approvata: dal PO

# Pianificazione dello sprint

- Il team sceglie dal product backlog gli elementi che verranno certamente realizzati nello sprint
- Studio dell'architettura di alto livello
- Si crea lo sprint backlog: i compiti da fare
  - Identificazione e stima di ciascun compito (1-16 hours)
  - Uso di Planning Poker

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)  
Code the user interface (4)  
Write test fixtures (4)  
Code the foo class (6)  
Update performance tests (4)

# Planning poker





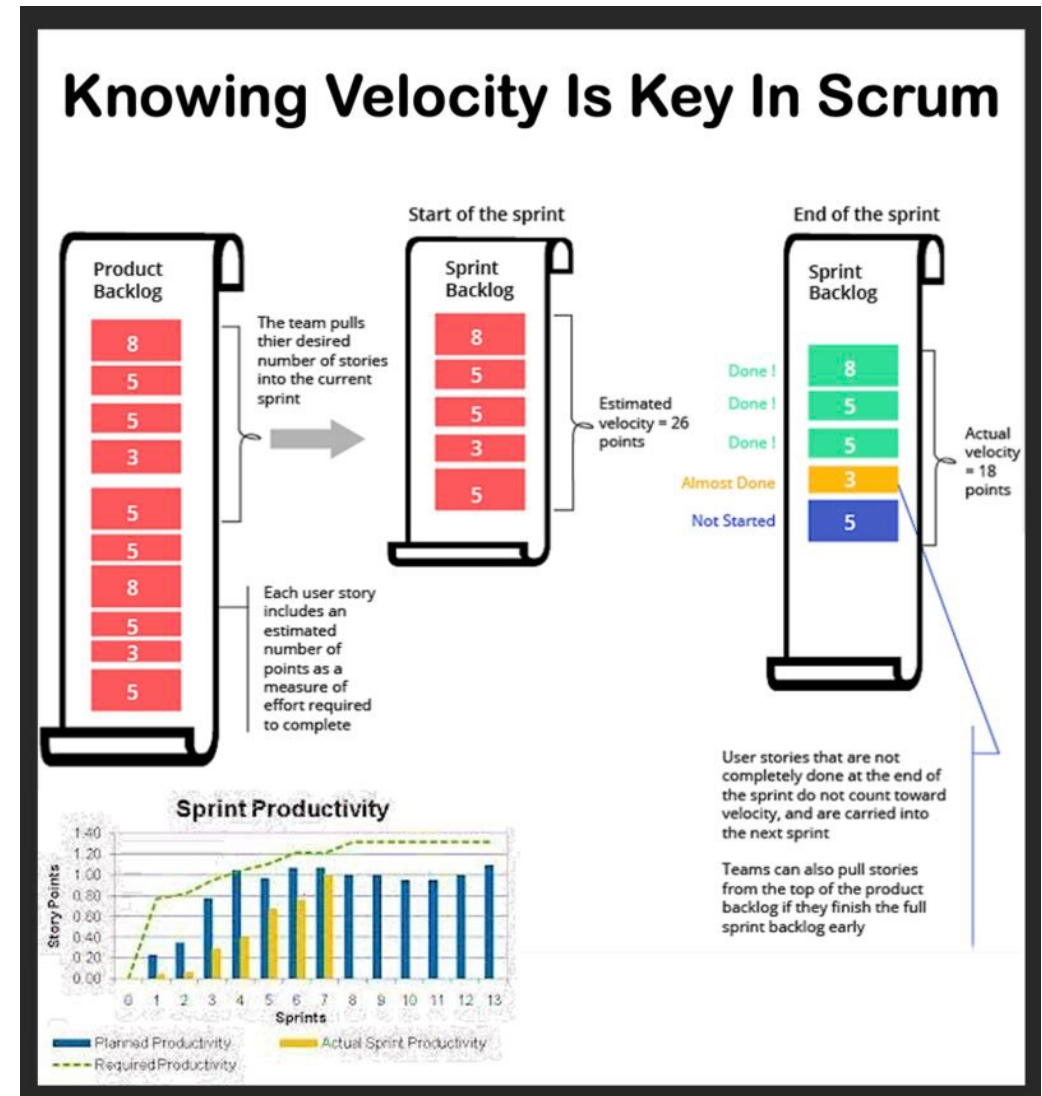
# Pianificazione dello sprint

La “velocità” è la quantità di story point che il team ha completato durante uno sprint

La velocità si calcola alla fine di uno sprint ed è utile per prevedere (stimare) quanto lavoro si riuscirà a fare nello sprint successive

La velocità si può misurare

- Per ciascun singolo task
- Per ciascuno sprint





# Scrum quotidiano

- Modalità
  - Guardando il board
  - 15-minuti max
  - In piedi
  - Chiunque può assistere
  - Possono parlare solo i pig: membri del team, ScrumMaster, Product Owner
  - Perché in piedi: evitare durata troppo lunga



## Daily scrum: Tre domande a tutti

1  
Che hai fatto ieri?

2  
Che farai oggi?

3  
Prevedi problemi?

Le risposte sono promesse ai compagni di team

# Il PO nel daily scrum

Il PO partecipa al meeting quotidiano, ma

- Non importa se il PO è assente (il meeting inizia comunque)
- Il PO non può inserire nuove US (questo si fa solo all'inizio dello sprint)
- Il PO non pretende di avere chiari gli stati di avanzamento: gli sviluppatori si autogestiscono
- Il PO partecipa per dare suggerimenti costruttivi

# (Scherzo)

How to make meetings shorter:



# Sprint review: la demo del prodotto

- Il team presenta al PO ciò che ha ottenuto con lo sprint: occorre mostrare come il prodotto è “cresciuto”
- È una demo delle nuove funzioni
- Serve per verificare lo stato del product backlog
- Riunione informale
  - 2 h di preparazione, niente slide
- Partecipa tutto il team
- Invitare tutti



# Sprint retrospective

- La retrospettiva riguarda il processo (non il prodotto): come sta andando lo sviluppo?
- Occorre analizzare periodicamente cosa va e cosa non va in ciascuno sprint
- Quando: alla fine di ciascuno sprint, dopo la review
- Partecipano tutti:
  - ScrumMaster (coordinatore della retrospettiva)
  - Product Owner
  - Team
  - Forse commitenti, utenti e altri stakeholders

# Retrospektiva: Start / Stop / Continue

Il team discute cosa vorrebbe fare:

Start doing

Stop doing

Continue doing

*Ci sono vari modi alternativi di definire una retrospektiva*

# Lo scopo della retrospettiva

- Esaminare come è andato l'ultimo sprint riguardo a persone, relazioni, processi e strumenti;
- Rivedere i principali driver architetturali rispetto alle attese degli stakeholder
- Identificare e ordinare gli elementi principali che sono andati bene e le migliorie potenziali;
- Creare un piano per attuare i miglioramenti al modo di lavorare del Team



# Cancellare uno sprint

- Il Product Owner può cancellare uno sprint durante il suo svolgimento se l'obiettivo dello sprint diventa obsoleto
  - per es. se sono cambiate le condizioni di mercato o se l'organizzazione subisce una modifica
- Di solito non ha senso cancellare uno sprint

# Scrum: gli artefatti

## Ruoli

- Product owner
- ScrumMaster
- Team

## Eventi

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

# Product backlog

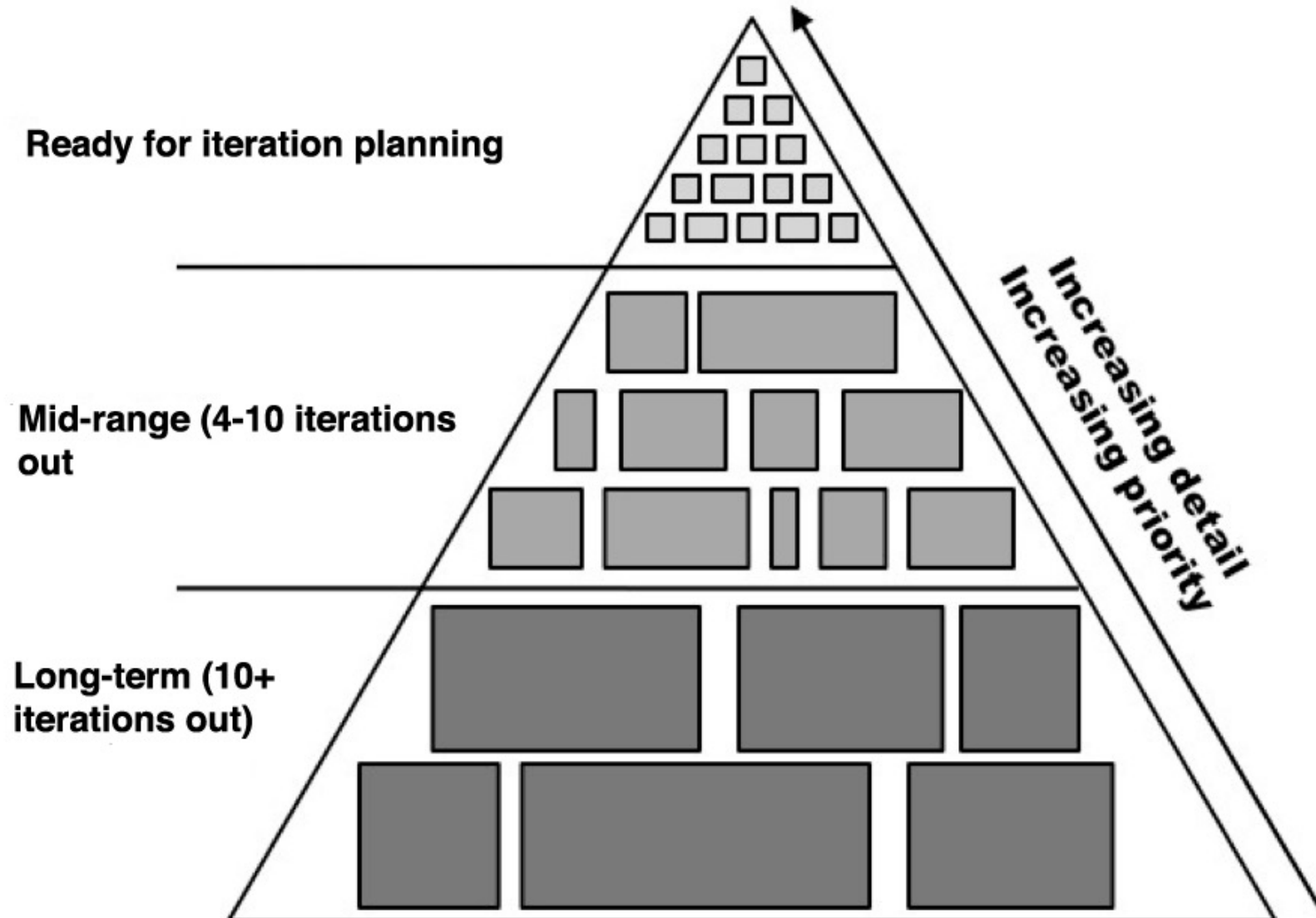


Questo è il  
product backlog

- Product backlog = Lista dei requisiti in forma di user stories
- Definita in modo tale che ciascun elemento abbia valore per gli utenti o i committenti del prodotto
- Messa in priorità dal Product Owner
- Priorità ridefinite all'inizio di ogni sprint

# La struttura del product backlog

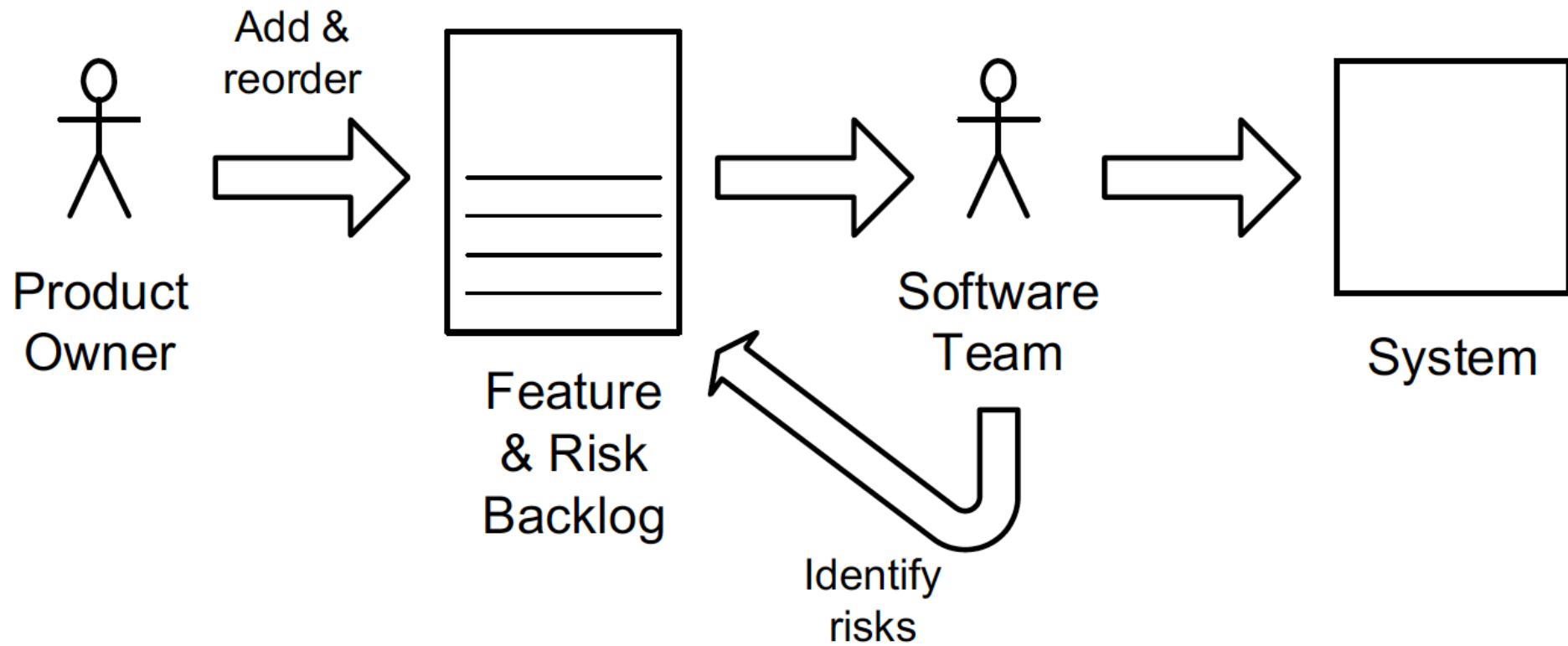
**Break down stories as they move up the backlog**



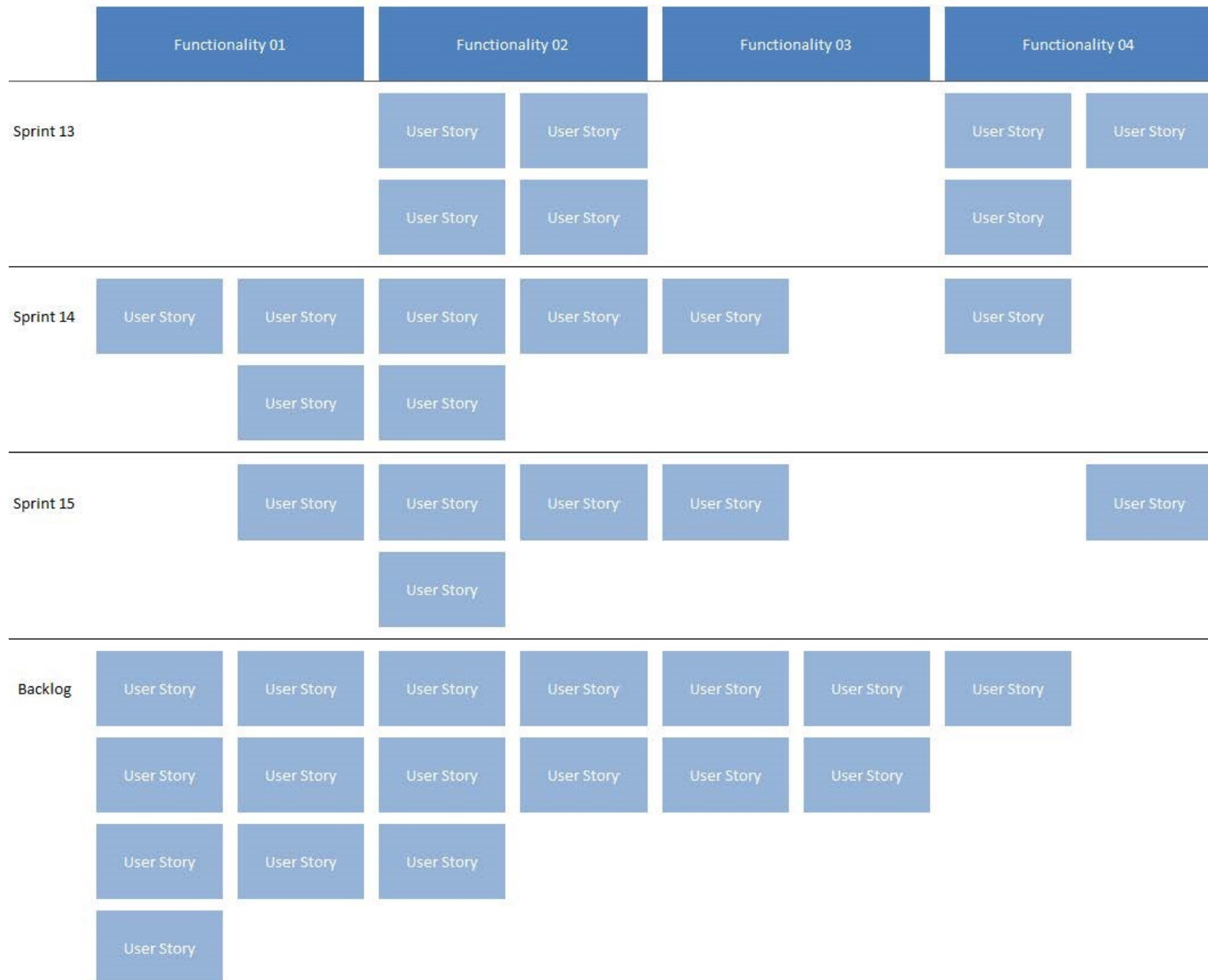
# Esempio di product backlog

Elemento backlog	Stima (h)
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

# Variante del product backlog con analisi dei rischi



# Mettere in priorità le user story

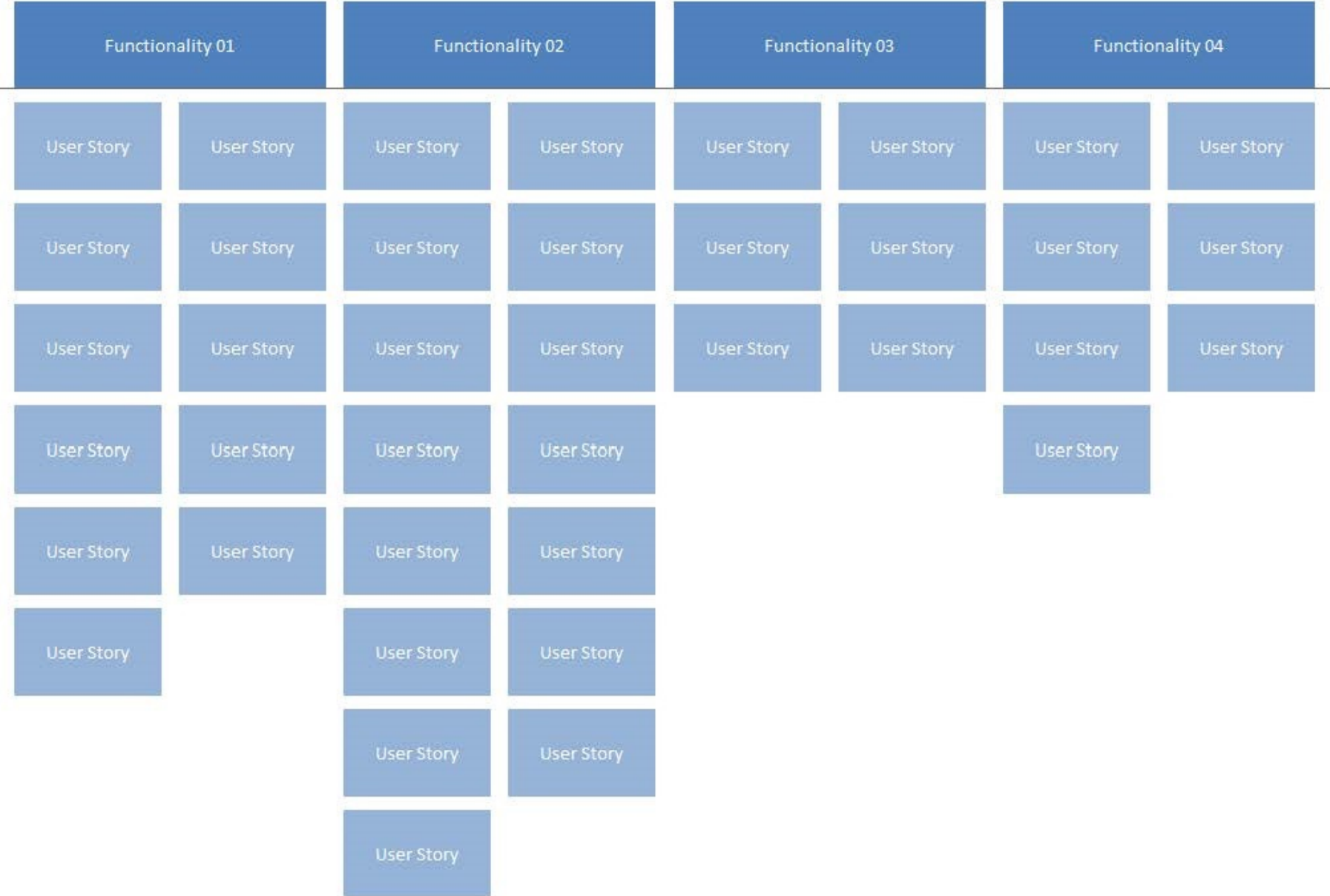


# User story map: come affettare il product backlog

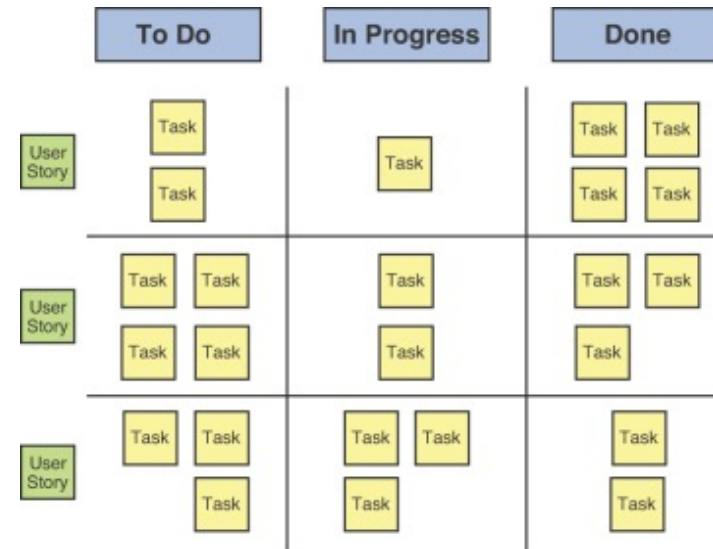
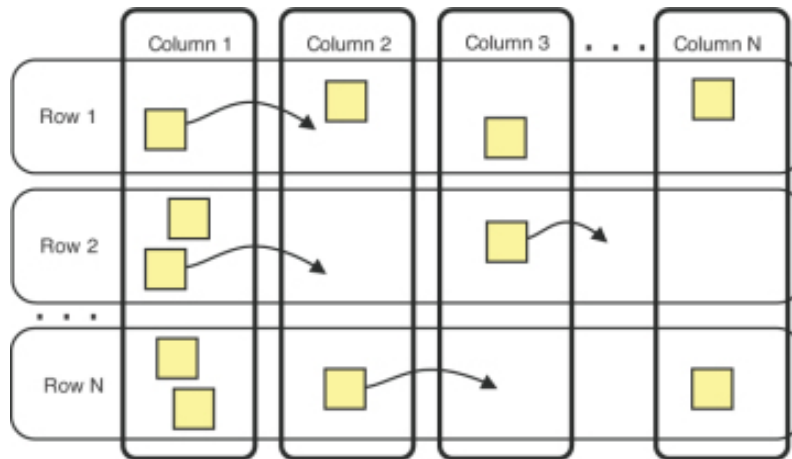




# Mappare le user story sulle feature



# Come si usa il backlog



# Taskboard

Agilar Taskboard

http://taskboard.agilar.org/taskboard/team/6

Smart Bookmarks - Google Translate - Gmail - Inbox - Google Calendar - Google Reader (100+) - Wikipedia - Plesk 8.1.0 - BIT-HDTV - Home o... - site.dev.agilar.org - Taskboard DEV - iLean DEV

Wikipedia, the free encyclop... - Scrum Alliance - Certified Sc... - Agilar Taskboard - Dashboard [Hudson] - RFQ detail - Gmail - Inbox - xquesada@g...

Product Backlog - Add Task

PLEASE ANALYSE - PLEASE TEST - DONE - BLOCKED - HIGH PRIORITY - WAITING - DELEGATED - BUG

FRANCISCO - XAVIER - MARCOS - MARCO

Product Backlog - Add Task - Product Backlog - Add Task

Agilar Dev Team

AA018 Legacy Issues And Unplanned Items Priority: 3000 Size: 0

AP005 Training section and module - dirt road Priority: 2885 Size: 5

	NOT STARTED	IN PROGRESS	FINISHED
<p>Fix Bubbles</p> <p>Deploy to staging should backup the production database and test a production deploy</p>	<p>HUC - PLEASE TEST - make deploys know about the status of the last good build - FRANCISCO</p> <p>Sponsorship listings. - BUG - MARCOS N</p>	<p>When clicking everything goes to the homepage</p> <p>Check alignment of forms</p>	
<p>Add tests for active resource (see ActiveResource::HttpMock)</p> <p>Add tests to agora training controller (upcoming and past)</p> <p>Add AJAX calendar to start/end date</p>	<p>Add variations on start / end date - MARCOS N</p> <p>Change URL de produccion: #course_name - MARCOS N</p> <p>Mail to trainer and organizer when new registration occurs - MARCOS N</p>	<p>Change Trainings heading to "Public Trainings"</p> <p>Add tobias training</p>	
<p>"Your trainings" should only be visible to co-trainer but not main trainer - BUG</p> <p>You can stop publishing your profile on the site even if you have published trainings - BUG</p> <p>Possible to put same co-trainer and co-trainer - BUG</p>	<p>Only trainers who have their profile on the Agilar website should be eligible for trainings - FRANCISCO</p> <p>BUG de trainings containing (website) - link apunta a... - FRANCISCO</p> <p>Support for external registration mechanism (URL field) - PLEASE TEST - MARCOS N</p>		
<p>The owner of a training should be able to edit and delete it</p>	<p>Cannot Edit existing trainings - PLEASE TEST - MARCOS N</p>		

Done

# Obiettivo dello sprint (sprint goal)

Breve descrizione del lavoro da fare durante lo sprint. Esempi:

## Database Application

Make the application run on SQL Server in addition to Oracle.

## Life Sciences

Support features necessary for population genetics studies.

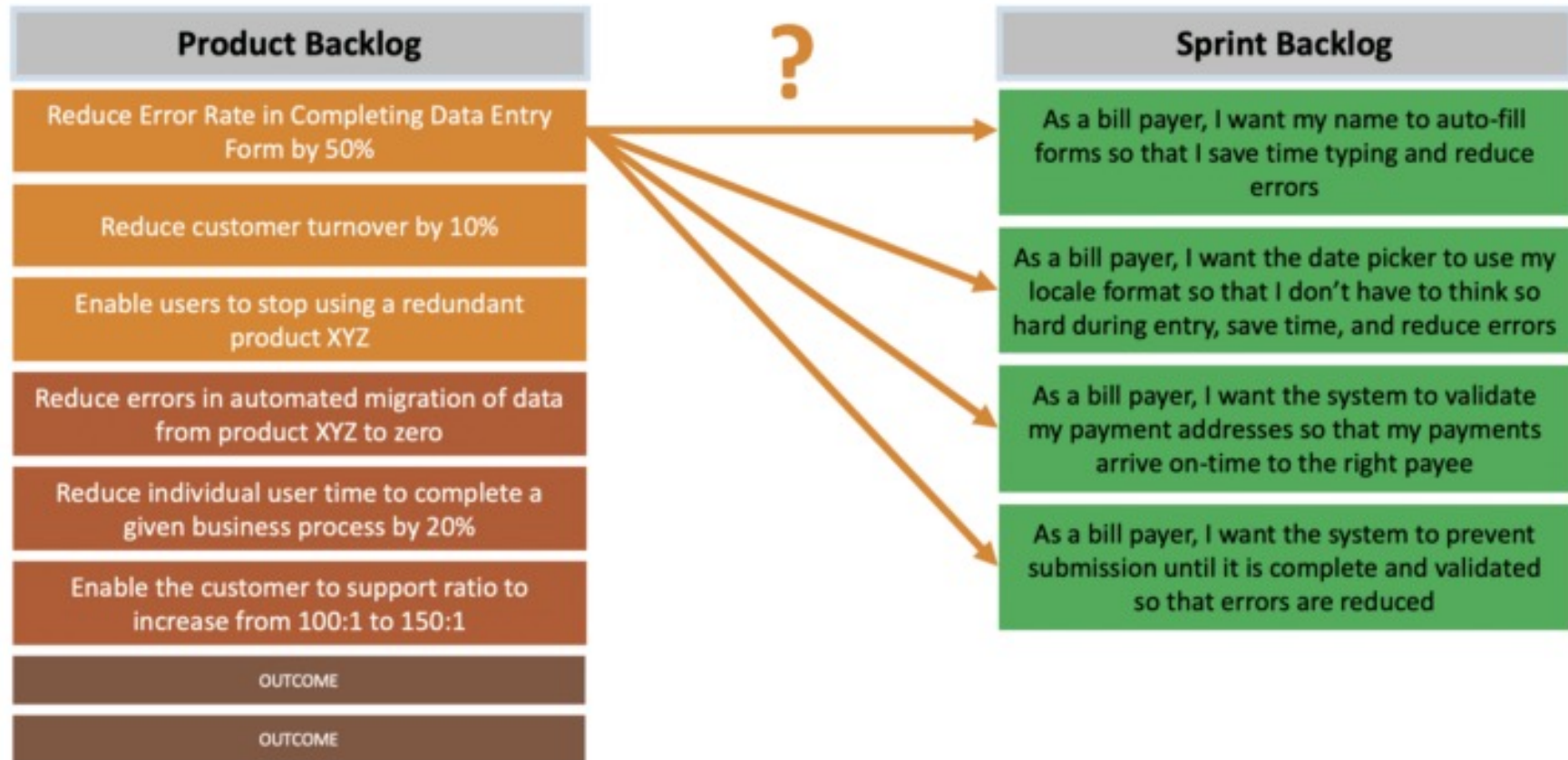
## Financial services

Support more technical indicators than company ABC with real-time, streaming data.

# Gestione dello sprint backlog

- I membri del team prenotano lavoro da fare su scelta personale
- Il lavoro non viene assegnato, ma richiesto su base “volontaria”
- La stima del lavoro da fare in termini di effort viene aggiornata quotidianamente

# Dal backlog di prodotto a quello di sprint



# La decomposizione delle US in task

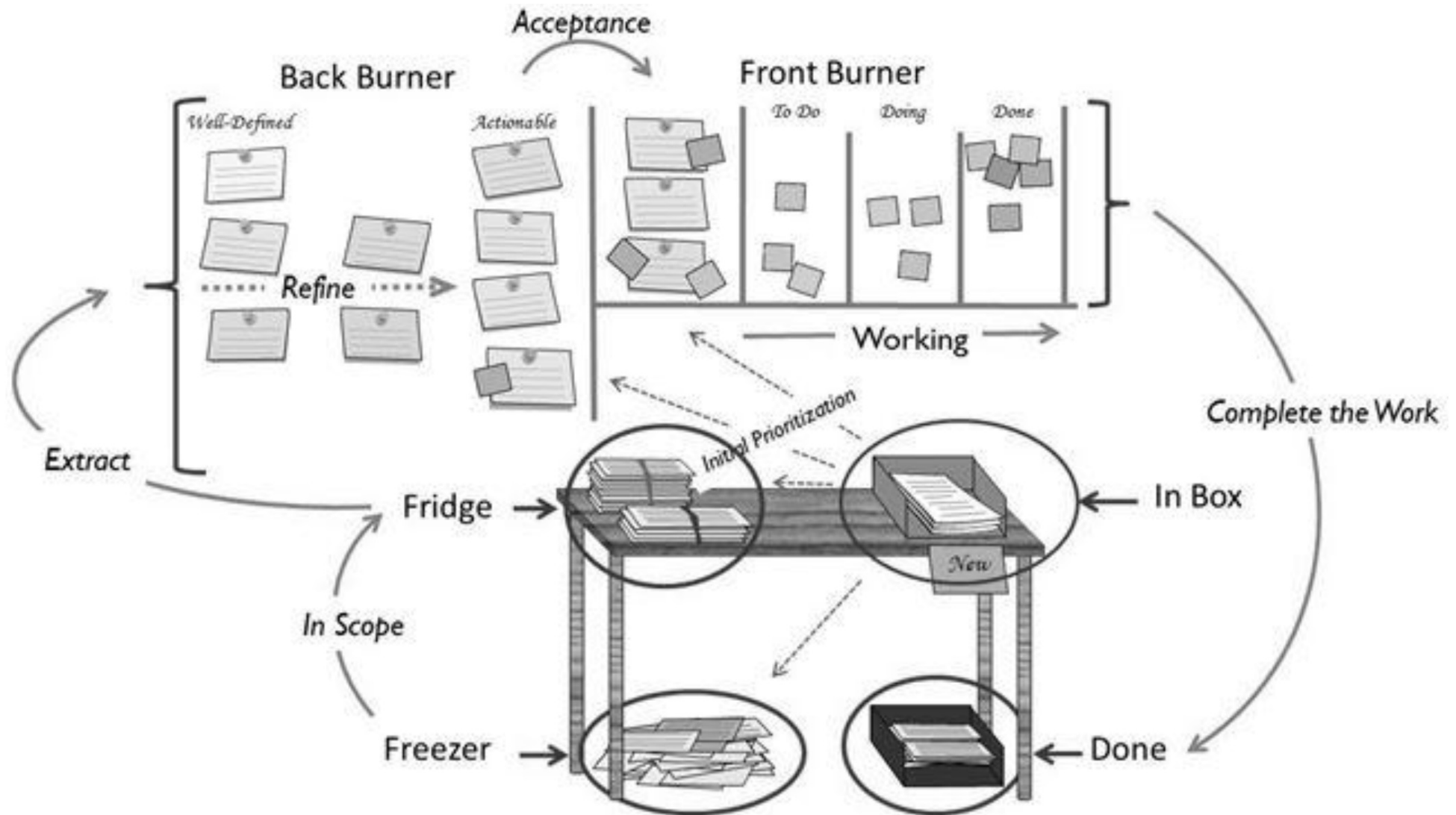
USER STORY	TASKS	DAY 1	DAY 2	DAY 3	DAY 4	DAY 5	...
<b>As a member, I can read profiles of other members so that I can find someone to date.</b>	Code the...	8	4	8	0		
	Design the...	16	12	10	4		
	Meet with Mary about...	8	16	16	11		
	Design the UI	12	6	0	0		
	Automate test...	4	4	1	0		
	Code the other...	8	8	8	8		
<b>As a member, I can update my billing information</b>	Update security tests	6	6	4	0		
	Design a solution to...	12	6	0	0		
	Write a test plan	8	8	4	0		
	Automate tests...	12	12	10	6		
	Code the...	8	8	8	4		

# Gestione dello sprint backlog

- Ogni membro del team può modificare lo sprint backlog, che di solito viene conservato in un tabellone detto “taskboard” o “kanban”
- Il lavoro da fare in ogni sprint “emerge” sul tabellone
- Se il lavoro da fare è poco chiaro, conviene definire uno sprint backlog item con una stima maggiore e decomporlo più tardi
- Occorre aggiornare il lavoro da fare man mano che viene fuori



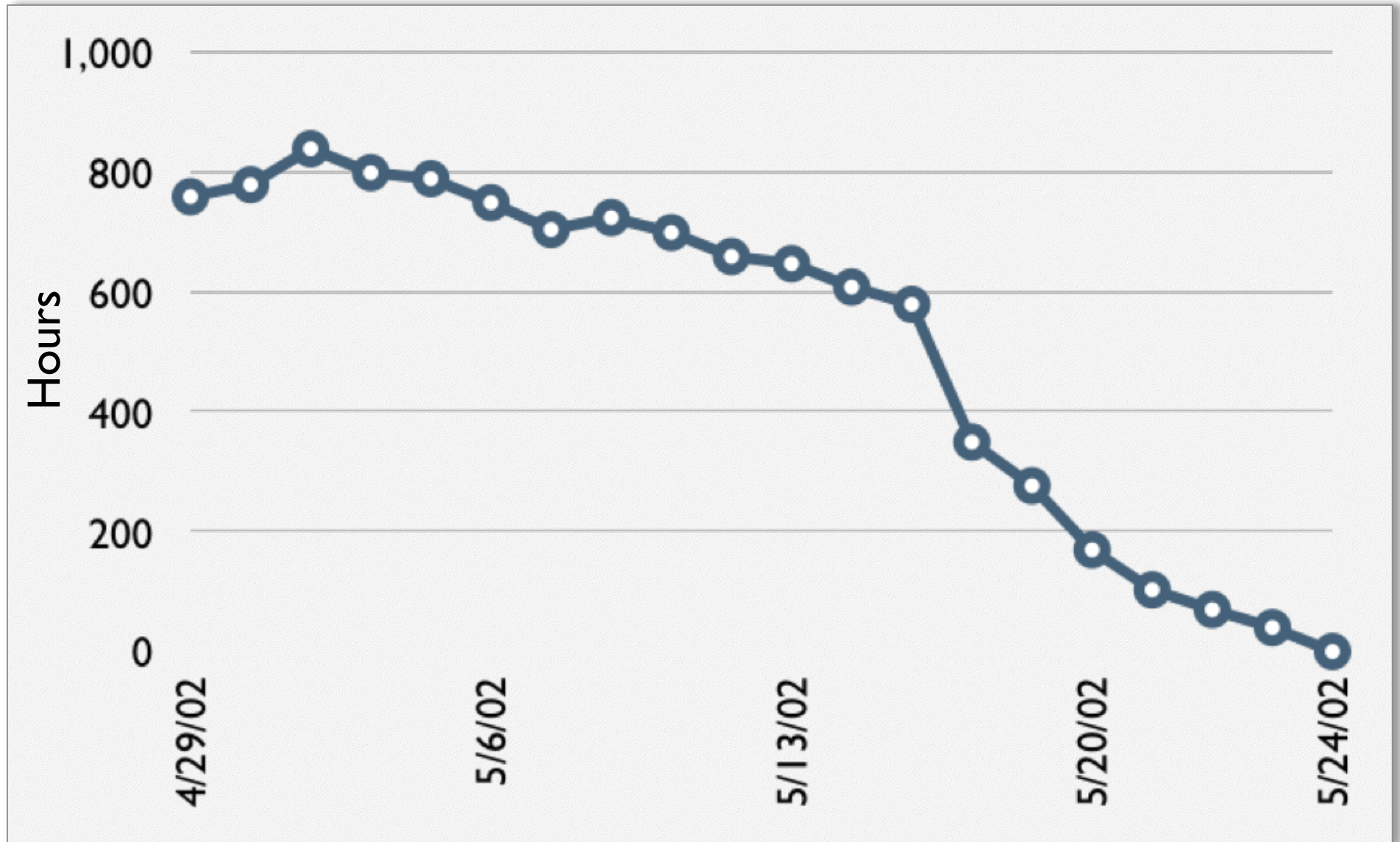
# Il ciclo dei compiti in uno sprint



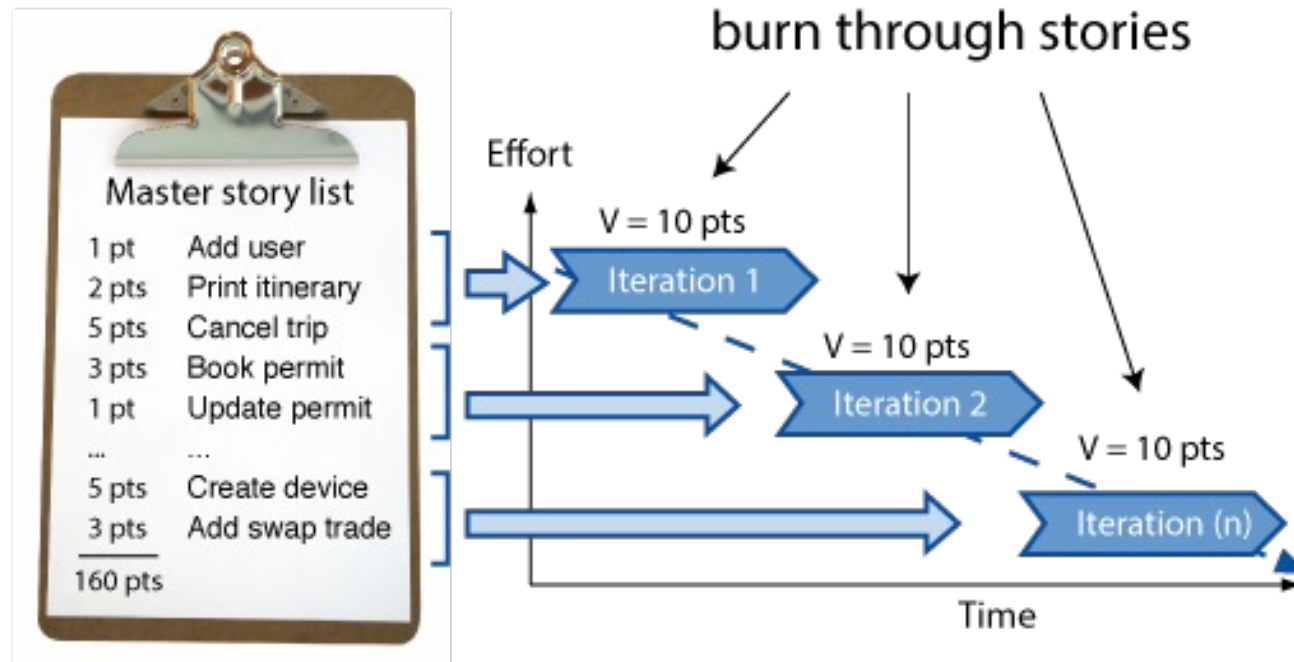
# Uno sprint backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

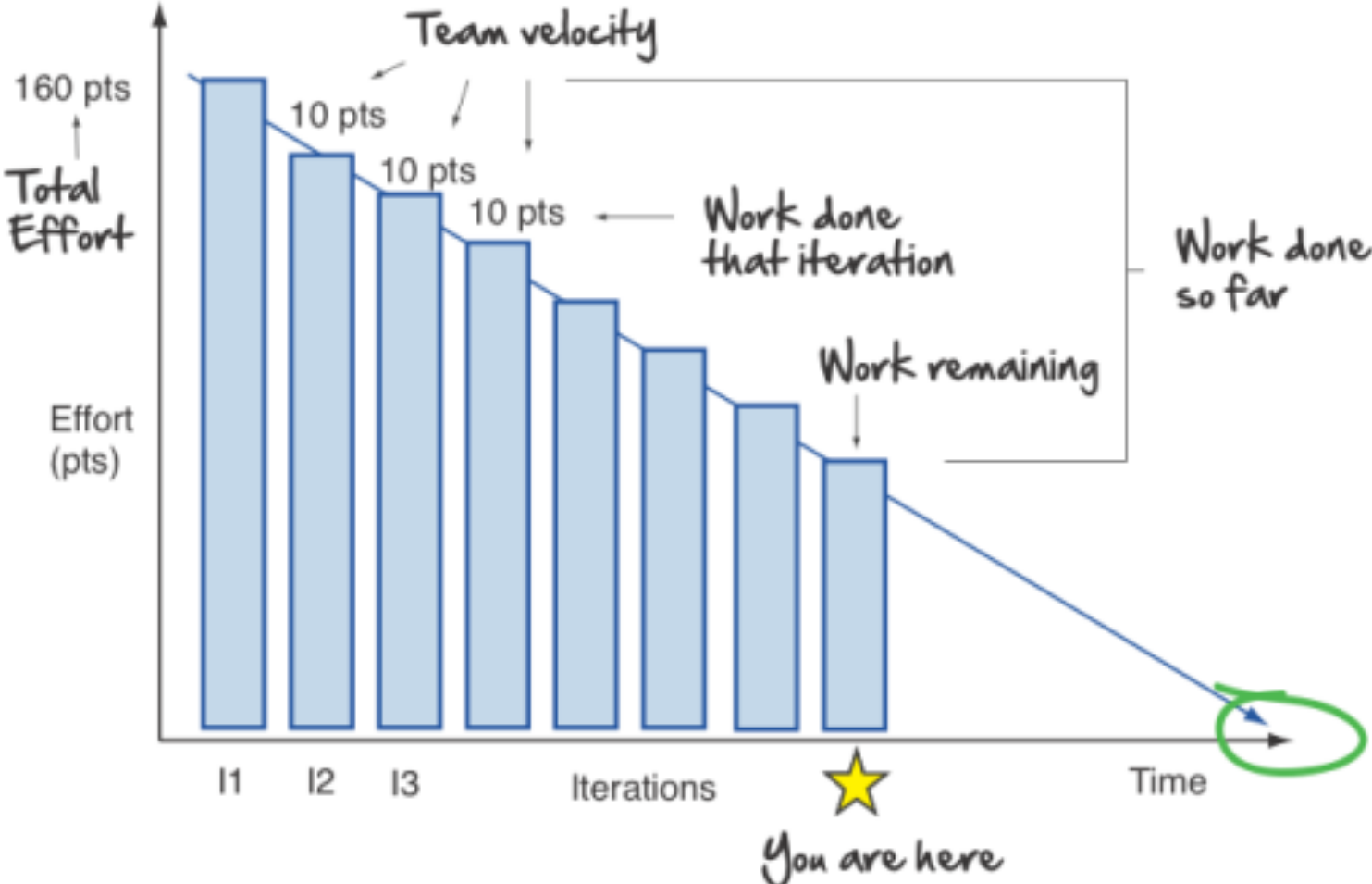
# Un burndown chart di uno sprint



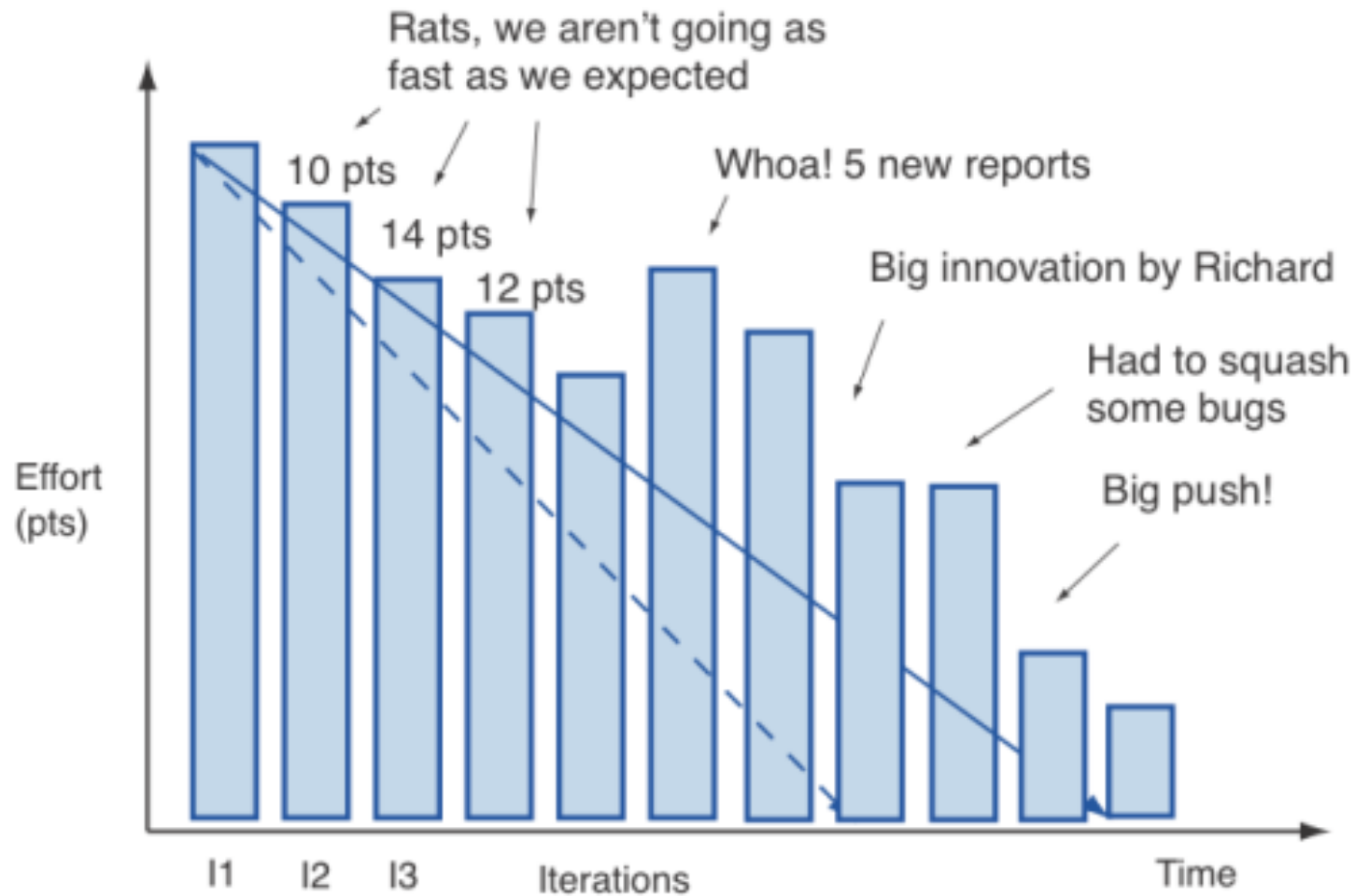
# Burndown charts



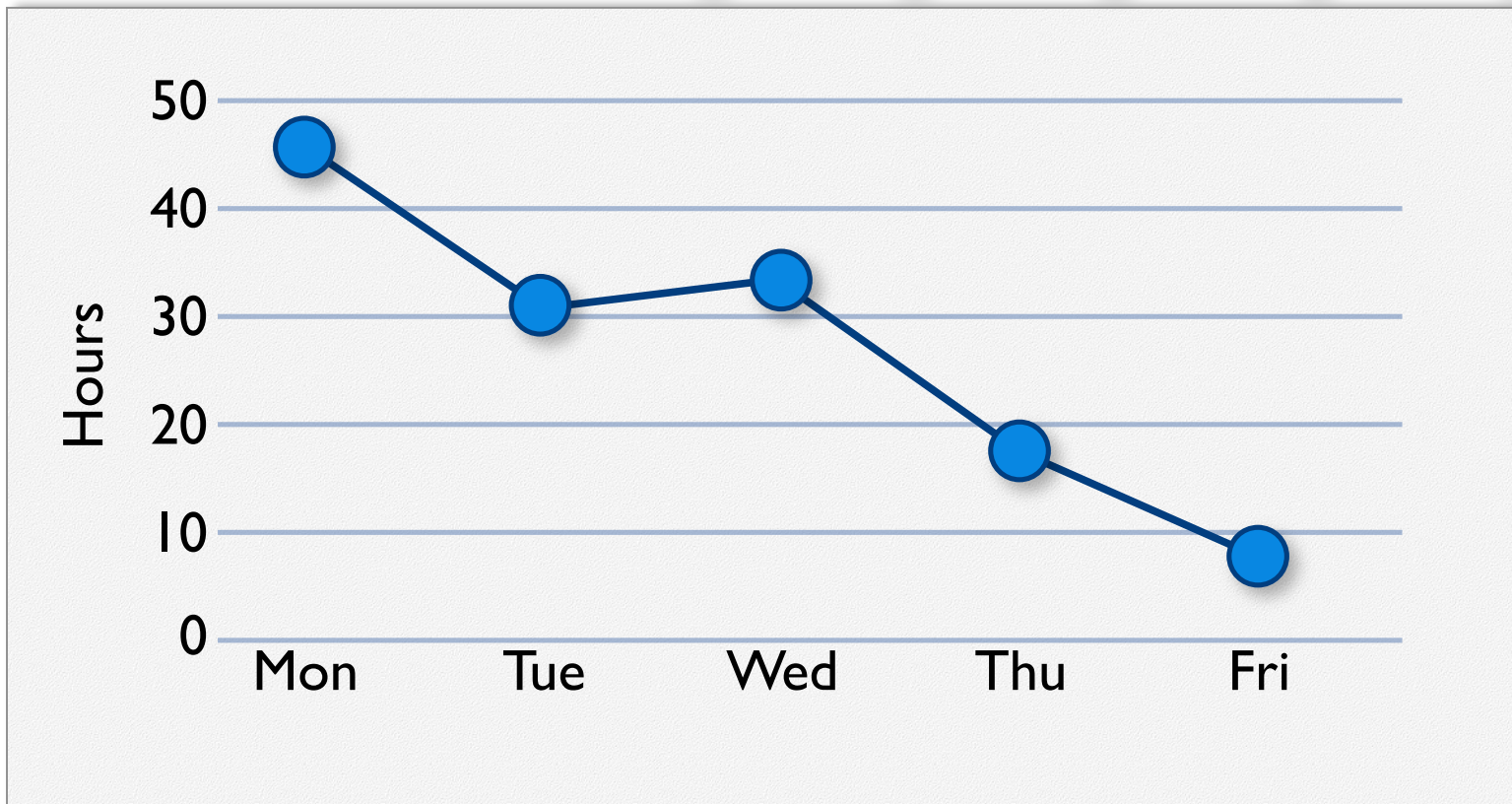
# Velocity ideale sul burndown chart



# Velocity reale sul burndown chart

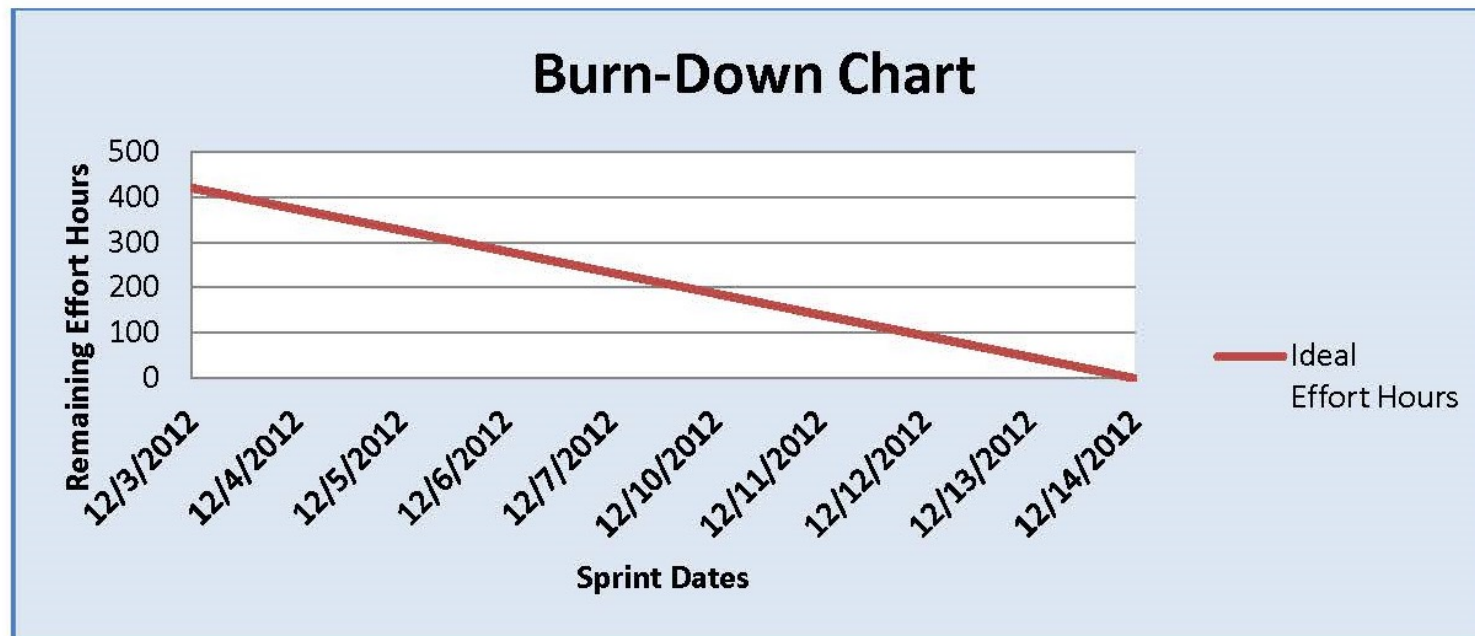


Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



# Esempio

Durata sprint: 2 settimane, 7 persone, 6h/giorno. Totale sforzo ideale 420 ore



<https://www.scrumalliance.org/community/articles/2013/august/burn-down-chart---an-effective-planning-and-tracki>



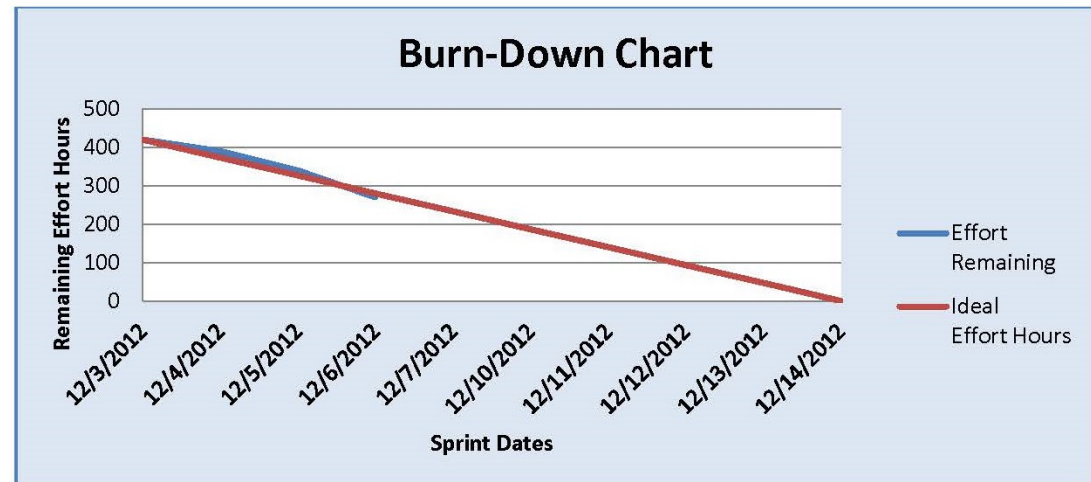
# Stima continua dello sforzo rimanente

Ogni partecipante sceglie un compito e stima il tempo rimanente.

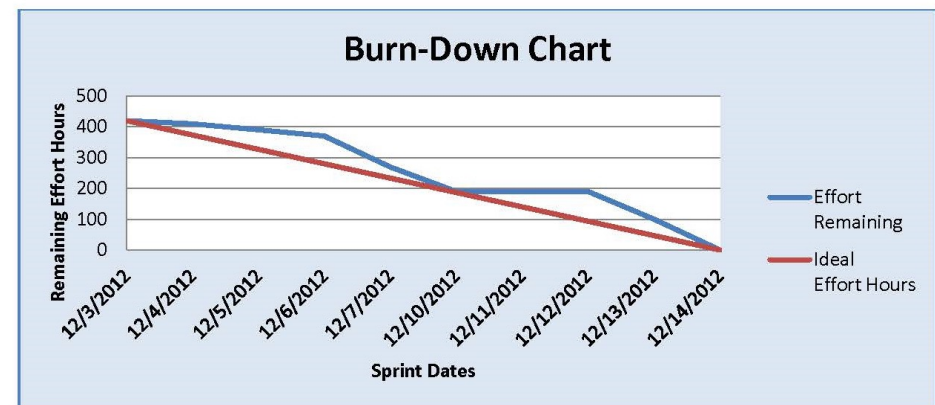
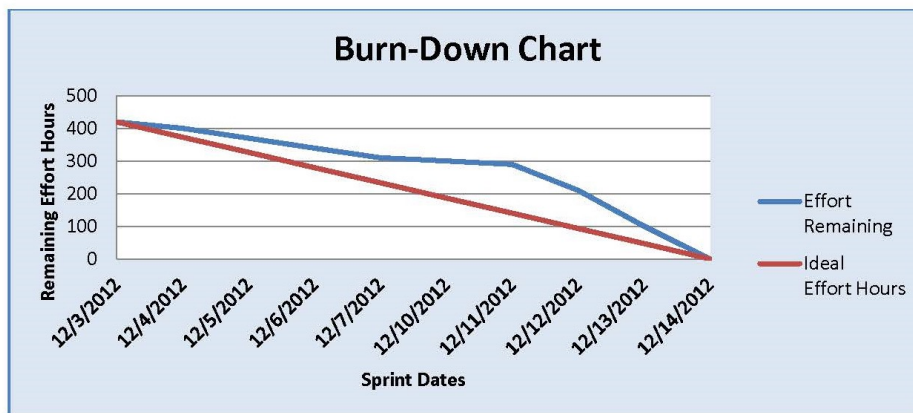
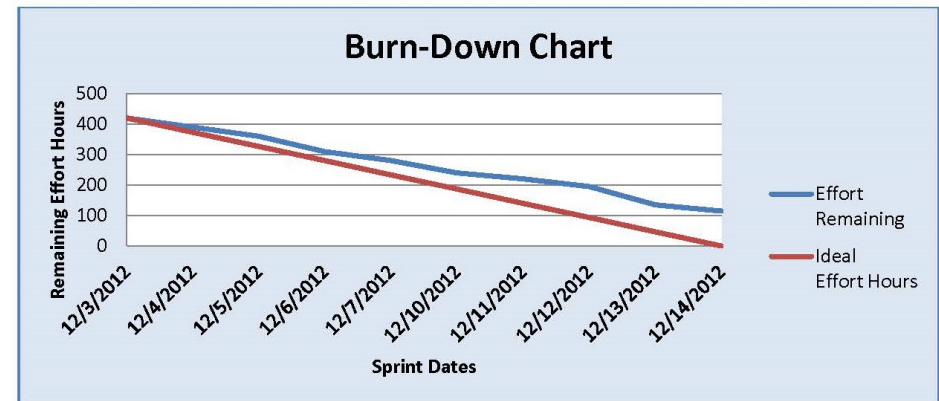
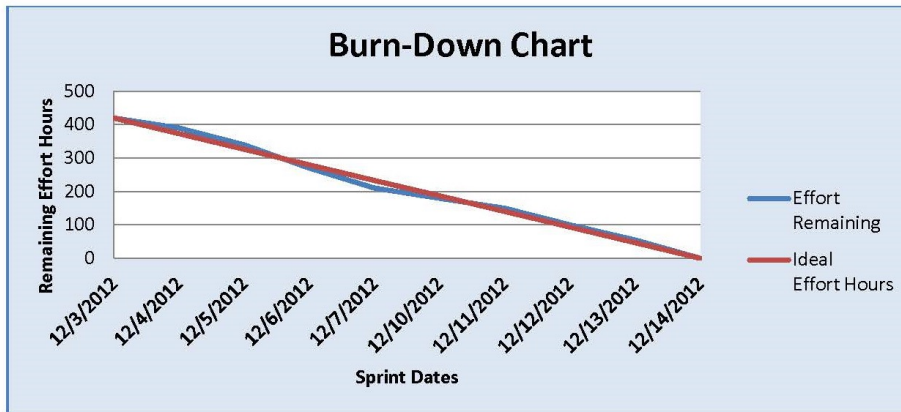
Per es. dopo il primo giorno sono state spese 6 h sul compito 1. Il programmatore stima che rimangano altre 6 ore (quindi 2h oltre la stima di 10h).

Il burn-down chart viene aggiornato e confrontato con l'ideale (blu vs rosso)

Story Name	Task No	Task Description	Status	Owner	Estimated Effort (in Hours)	Effort Remaining (in Hours)
Story 1	1	POC for Story 1	In Progress	Developer 1	10	6
	2	Requirement Clarification with PO	Closed	BA	8	0
	3	Develop modules	Open	Developer2	12	12



# Varie situazioni

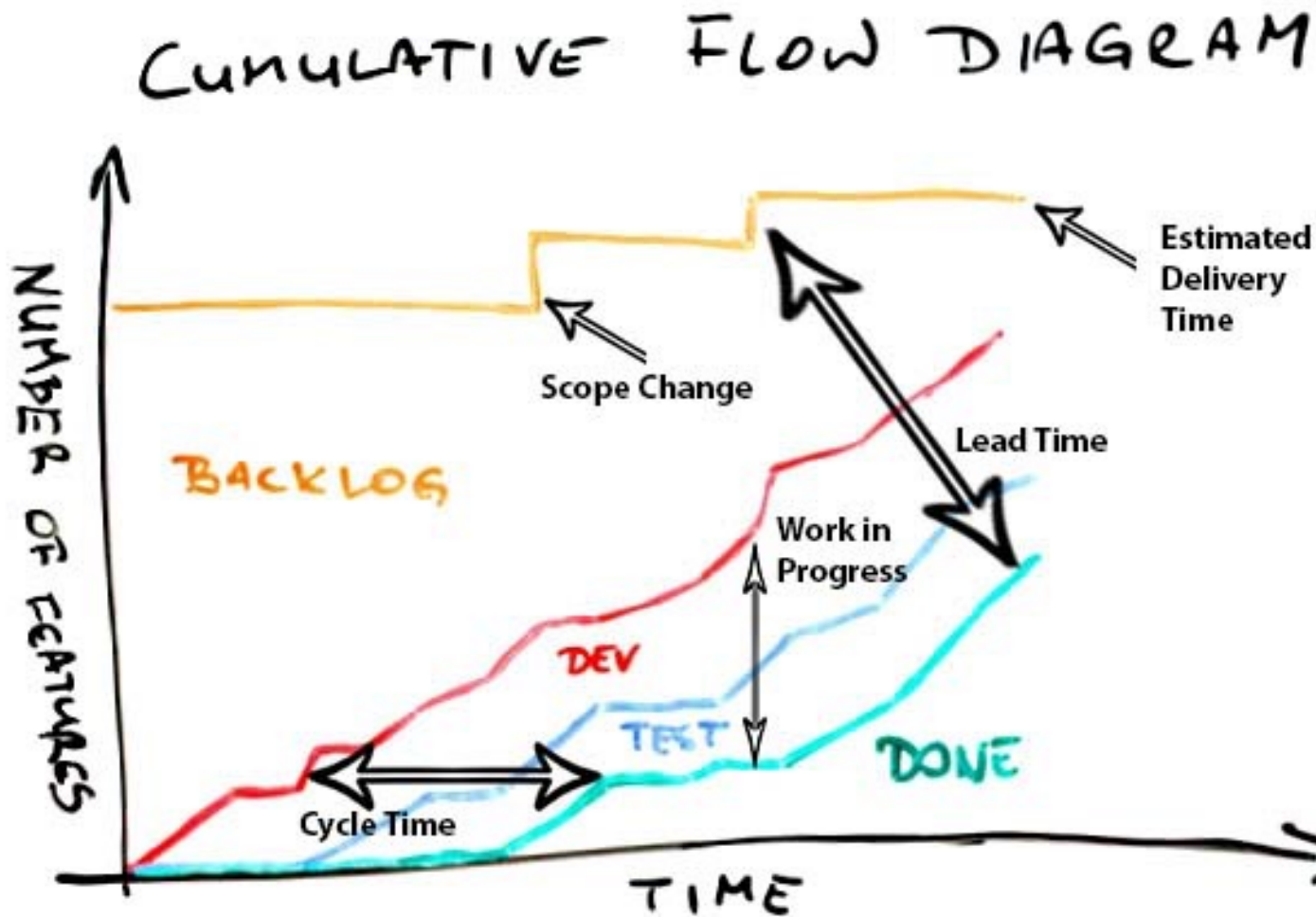


# Altre metriche agili

Oltre burndown e velocity, sono rilevanti:

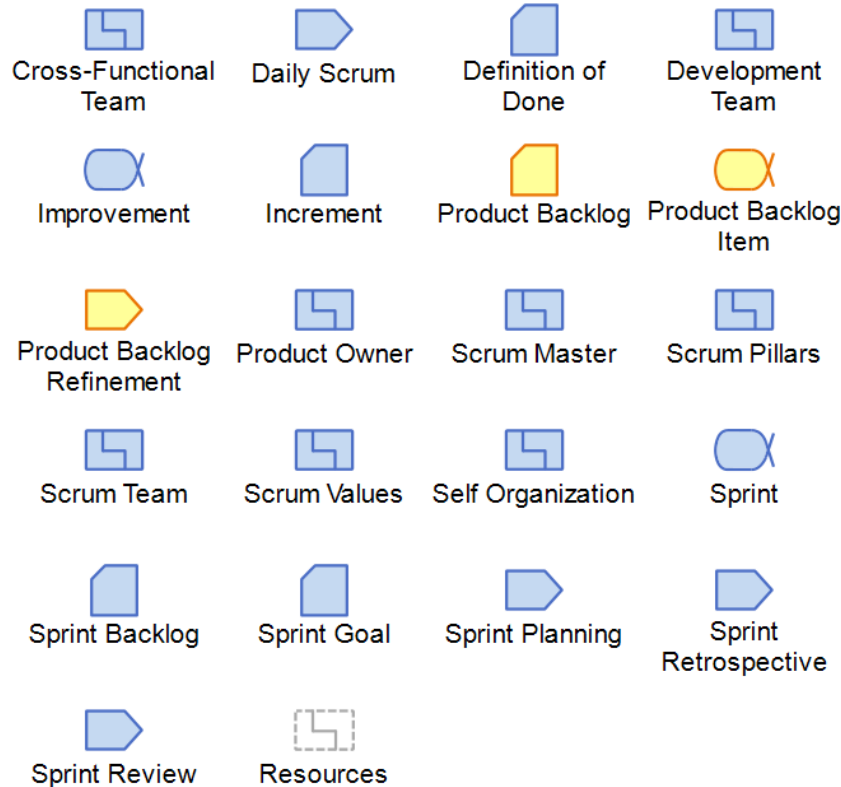
- Release burndown
- Control chart
- Cumulative flow diagram

# Cumulative flow diagram



# Scrum Essentials

Scrum is a framework for developing, delivering, and sustaining complex products.

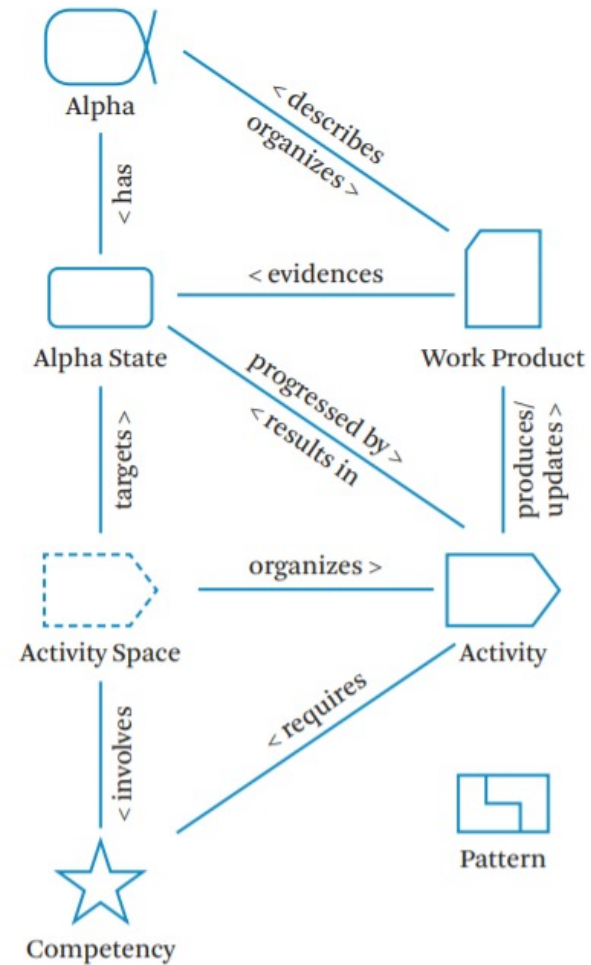
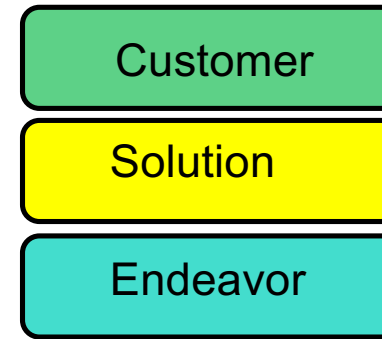


IVAR JACOBSON  
INTERNATIONAL

Generated by IJI Practice Workbench™

scruminc.

2.04



Legenda dei simboli Essence

# Varianti di Scrum

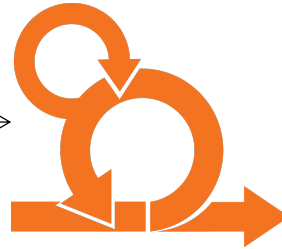
# Scrum multipli

- Alcuni progetti Scrum hanno coinvolto oltre 500 persone
  - Si scala su grossi sistemi con scrum di scrum
- Fattori che guidano la strutturazione del processo con Scrum multipli
  - Tipo dell'applicazione
  - Durata del progetto
  - Dispersione dei team

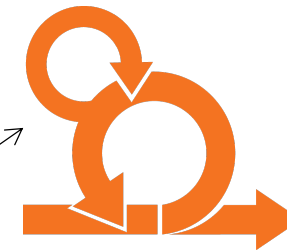
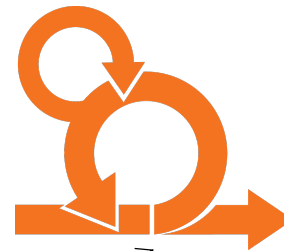
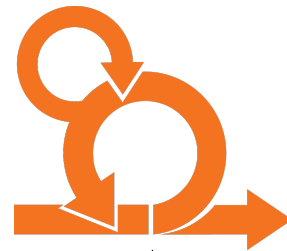
# Coordinare team multipli

Product backlog iniziale

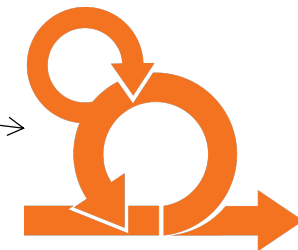
- Requisito funzionale
- Requisiti non funzionali
- Requisiti di scalabilità
- Altri requisiti funzionali e non



Team unico



Team multipli

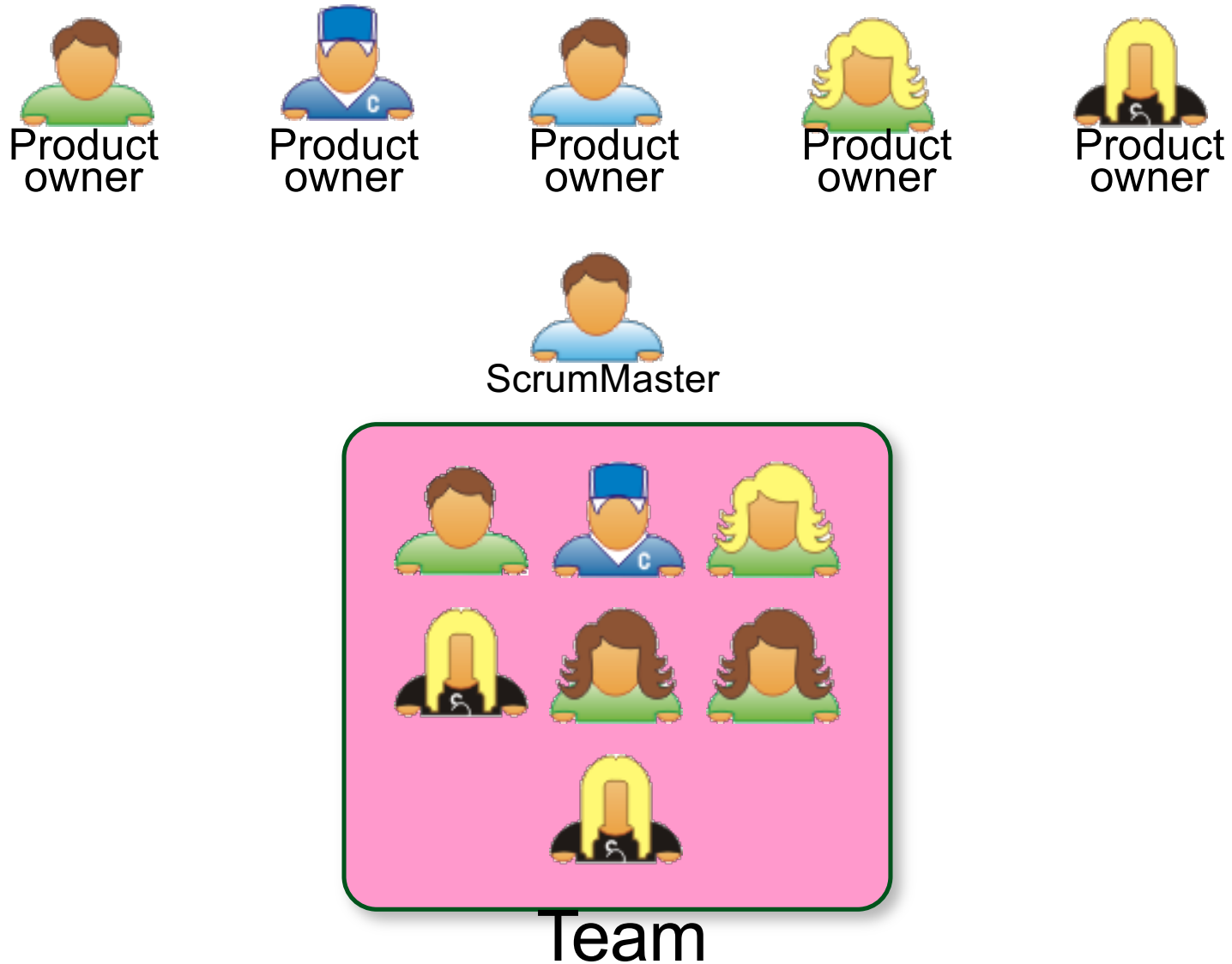


Product backlog

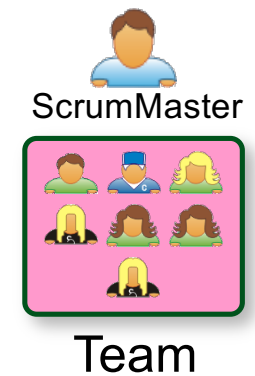
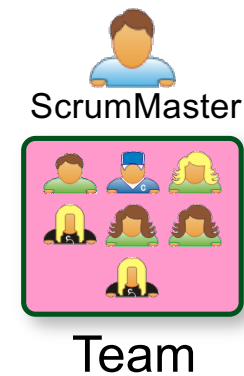
- Requisito funzionale
- Requisiti non funzionali



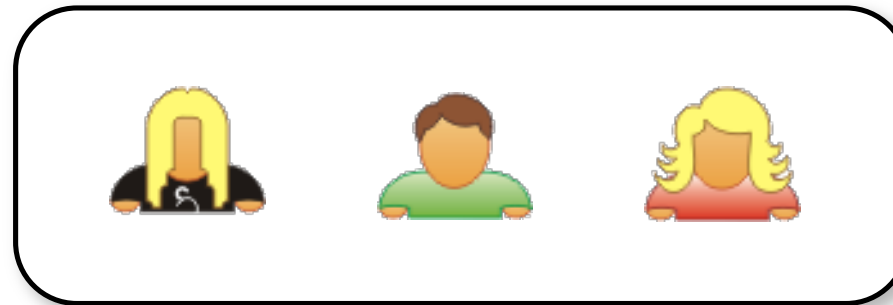
# Team unico, molteplici product owner



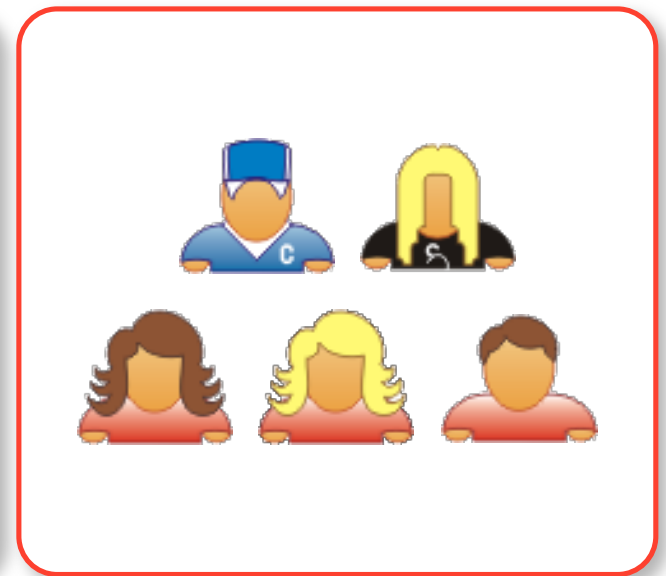
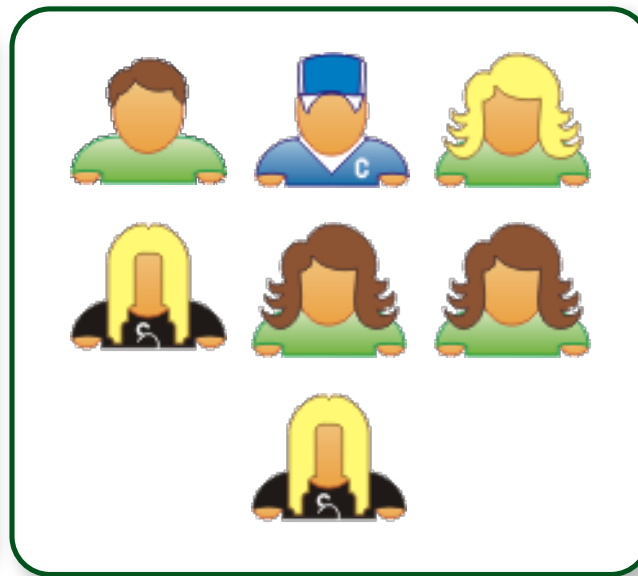
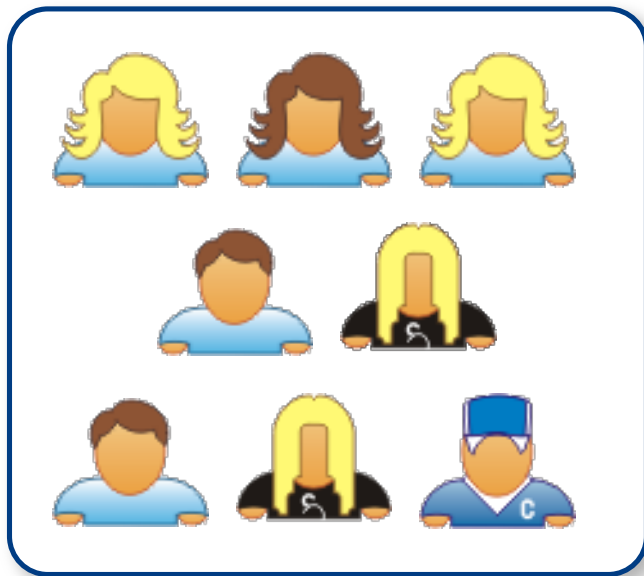
# Molteplici team, prodotto unico



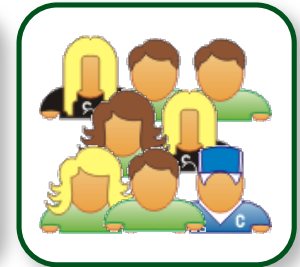
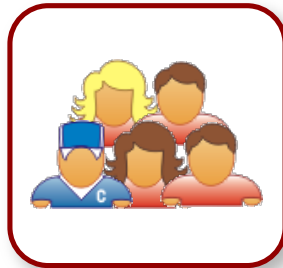
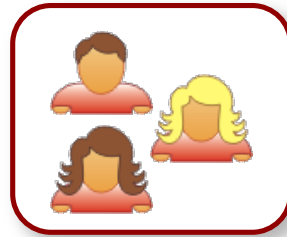
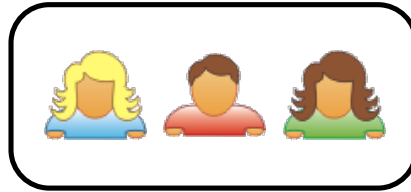
# Scrum di scrum



Metascrum  
degli ambasciatori



# Scrum di scrum di scrum



# J. Sutherland sullo scrum di scrum

- *Since I originally defined the Scrum of Scrum (Ken Schwaber was at IDX working with me), I can definitively say the Scrum of Scrums is not a "meta Scrum."*
- *The Scrum of Scrums as I have used it is responsible for delivering the working software of all teams to the definition of Done at the end of the Sprint, or for releases during the sprint.*
- *PatientKeeper delivered to production four times per Sprint. Ancestry.com delivers to production 220 times per two week sprint. Hubspot delivers live software 100-300 times a day.*
- *The Scrum of Scrums Master is held accountable for making this work*
- *So the Scrum of Scrums is a operational delivery mechanism.*

# Problemi tipici con Scrum

1. Ignoranza dei valori agili e di Scrum
2. Prodotto software non testato alla fine dello sprint (cattiva definizione di *“Fatto”*)
3. Backlog non pronto all’inizio dello sprint (cattiva definizione di *Ready*)
4. Mancanza di facilitazione (o cattiva facilitazione)
5. Mancanza di supporto da parte dei manager
6. Mancanza di supporto da parte degli stakeholder
7. Gestione caotica degli scrum di scrums

# Scrum: patti tra stakeholder e team

## **Il Team promette agli Stakeholder:**

- Il PO nel Team difenderà gli interessi degli Stakeholder
- Il tempo degli Stakeholder sarà usato saggiamente, verranno poste solo questioni utili allo sviluppo
- Il lavoro sarà della massima qualità, compatibilmente coi vincoli posti dall'azienda
- Alla fine di ogni sprint verranno consegnate nuove funzionalità che potranno essere validate dagli Stakeholders

## **Gli Stakeholder promettono al Team:**

- Gli stakeholders saranno disponibili per aiutare il Team quando necessario
- Lo ScrumMaster verrà coadiuvato a rimuovere gli ostacoli
- Priorità e vincoli non cambieranno durante uno Sprint senza il consenso del Team
- Partecipare ad uno sviluppo Scrum non danneggerà la carriera dei membri del Team

# XP o Scrum?

<b>XP</b>	<b>Scrum</b>
Orientato alla qualità (test driven)	Orientato al project management
Iterazione: 1-2 settimane	Sprint: 2-4 settimane
Requisiti sempre modificabili	Req modificabili alla fine dello sprint
Il cliente ordina le storie	Il team ordina le storie
Coaching informale	Scrum master certificato
Buone pratiche tipiche di XP: TDD, Pair programming, planning game, refactoring	Buone pratiche tipiche di Scrum: Retrospettiva post-mortem, uso di strumenti di PM, planning poker

**Nota bene: XP e Scrum possono coesistere**



# Riferimenti utili

- [www.scrum.org](http://www.scrum.org)
- [www.scrumalliance.org](http://www.scrumalliance.org)
- [scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Italian.pdf](http://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Italian.pdf)
- [scrumprimer.org/primers/it\\_scrumprimer20.pdf](http://scrumprimer.org/primers/it_scrumprimer20.pdf)
- [www.mountangoatsoftware.com/scrum](http://www.mountangoatsoftware.com/scrum)
- [www.controlchaos.com](http://www.controlchaos.com)

# Una lista di libri su Scrum

- Larman: *Agile and Iterative Development: A Manager's Guide*
- Cohn: *Agile Estimating and Planning*
- Cohn: *Succeeding with Agile*
- Cohn: *User Stories Applied for Agile Software Development*
- Derby & Larsen: *Agile Retrospectives*
- Highsmith: *Agile Software Development Ecosystems*
- Rubin: *Essential Scrum*
- Schwaber & Beedle: *Agile Software Development with Scrum*
- Schwaber: *Scrum and The Enterprise*
- Schwaber: *Agile Project Management with Scrum*

# Domande?

