

# Introduzione a Scrum



*Prof. Paolo Ciancarini  
Corso di Ingegneria del Software  
CdL Informatica  
Università di Bologna*

# Agenda

- Origini di Scrum: collaborazione responsabile
- Struttura dello sprint
- Ruoli
- Rituali
- Artefatti

# Manifesto: valori “agili”

Individui e interazioni

meglio  
di

Software che  
funziona

meglio  
di

Collaborazione  
del cliente

meglio  
di

Rispondere al  
cambiamento

meglio di

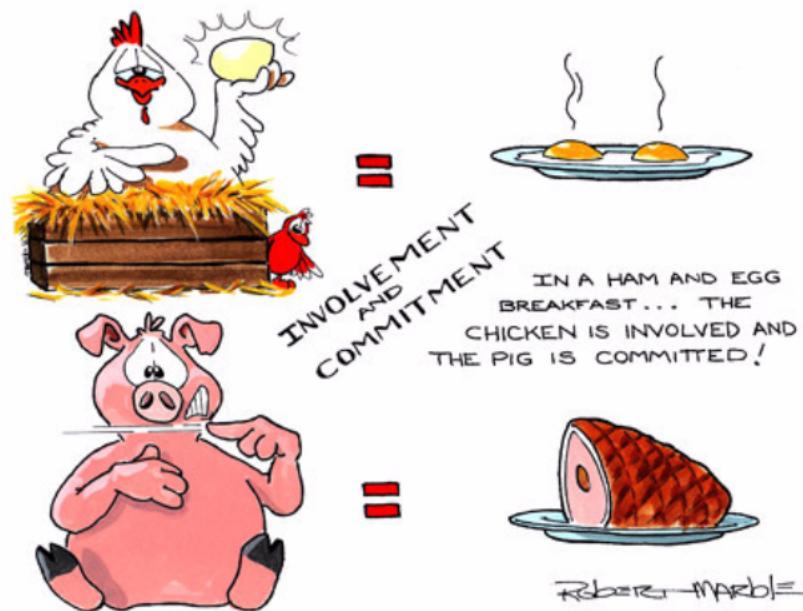
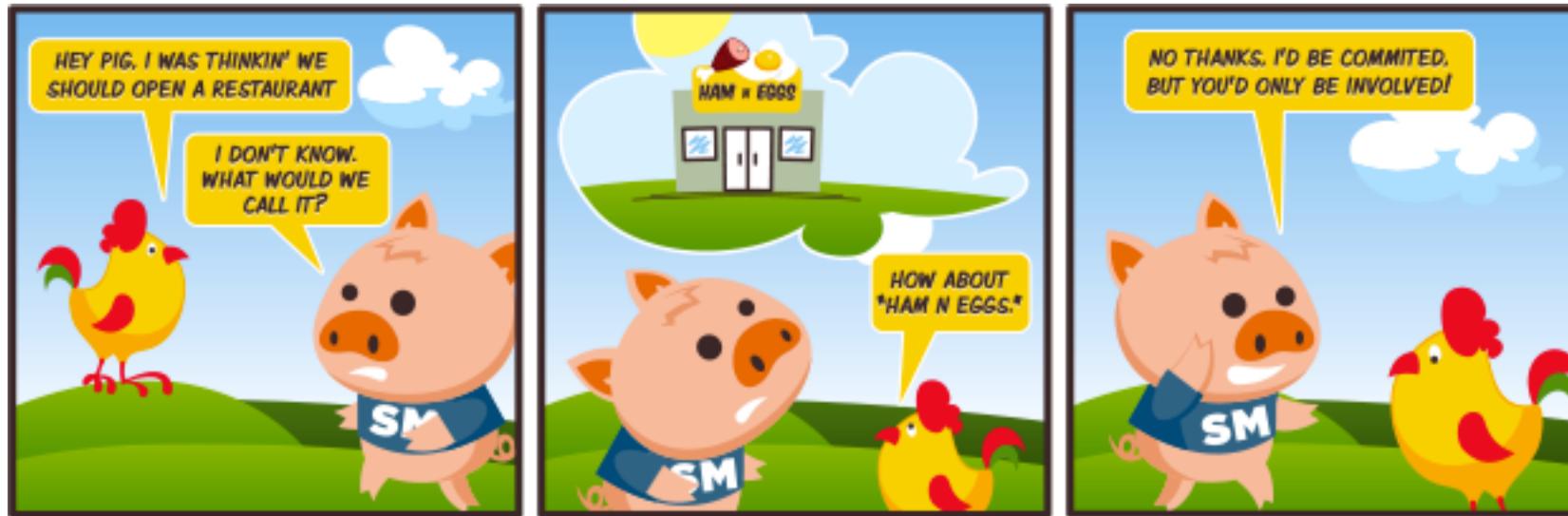
Processi e strumenti

Documentazione  
completa

Negoziazione  
contratto

Seguire un piano

# Galline e maiali



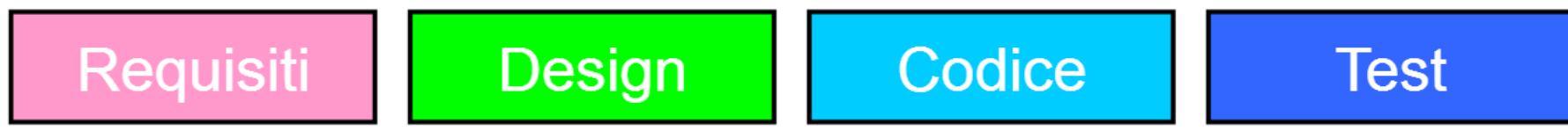
# Staffetta o pacchetto di mischia?

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.

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

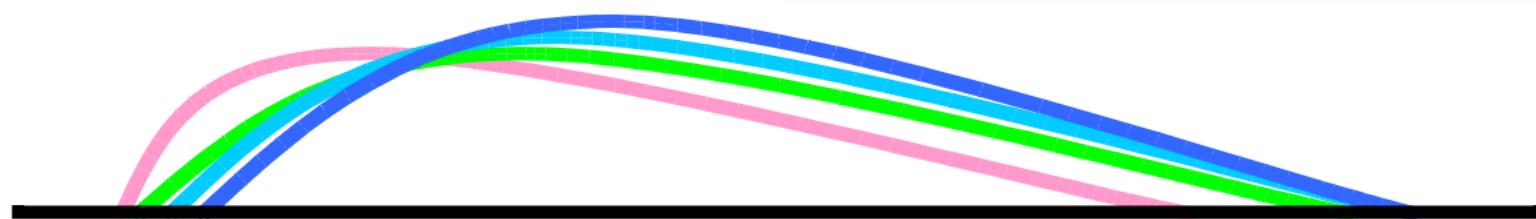


# Dalla staffetta al pacchetto di mischia



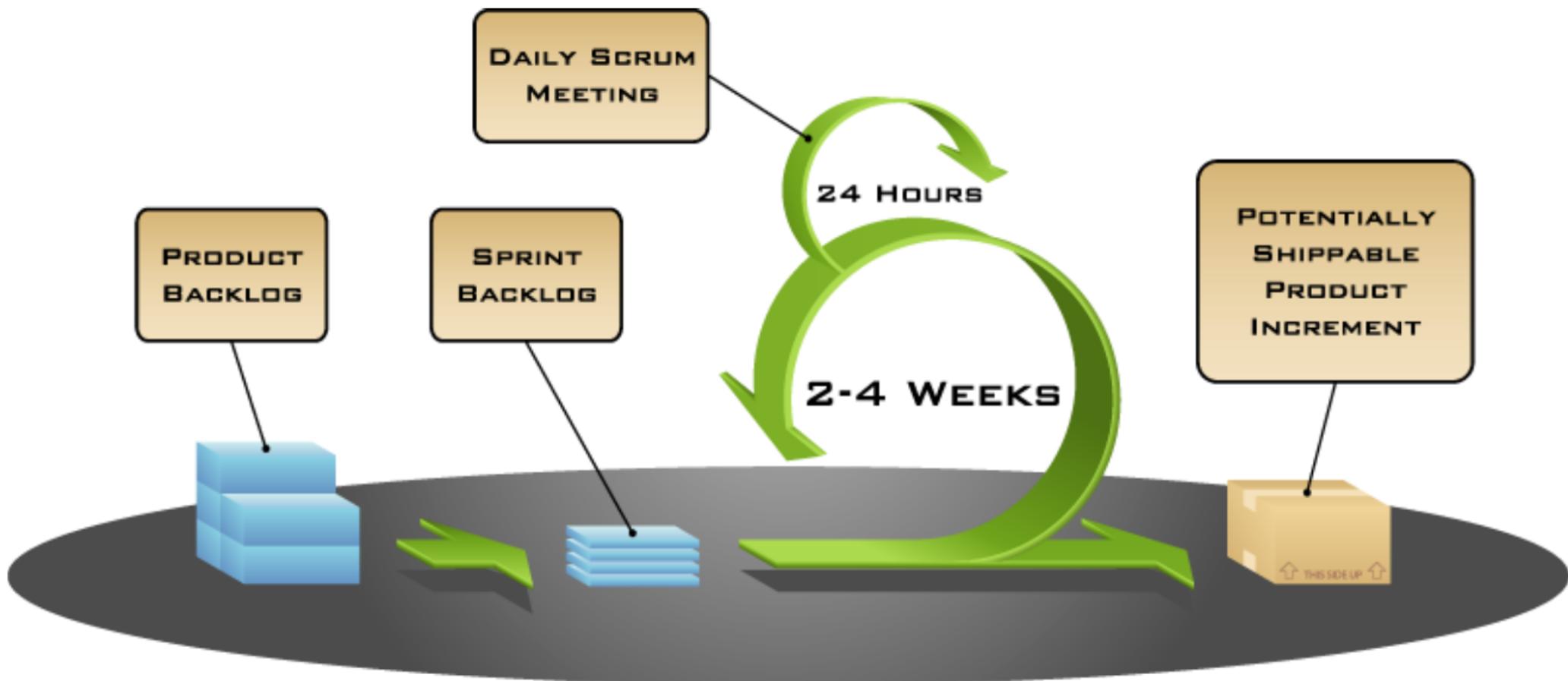
Non una cosa alla volta

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.

# Il modello Scrum



## Scrum in 100 parole

- Scrum è un modello di processo per produrre software ottenendo il massimo valore utile nel minor tempo
- Permette al cliente di ispezionare rapidamente e ripetutamente ogni 3-4 settimane versioni funzionanti del software
- Il cliente definisce funzioni e priorità. Il team di sviluppo decide il modo migliore di produrre le funzioni di più alta priorità.
- Ogni 3-4 settimane nasce una nuova versione che viene esaminata per decidere se continuare lo sviluppo con un altro sprint o produrne un rilascio

# Trasparenza

- Gli aspetti significativi del processo devono essere **visibili** ai responsabili del risultato finale (i “*pigs*”: il team, lo ScrumMaster, il Product Owner)
- La trasparenza richiede che tali aspetti siano definiti in modo tale che gli osservatori condividano una comune comprensione di ciò che viene visto

Esempi:

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

# Origini di Scrum

- Jeff Sutherland
  - 1993 Easel Corp
  - 500+ persone usavano Scrum
- Ken Schwaber
  - Articolo con Sutherland @OOPSLA 95
  - Autore di 3 libri su Scrum
- Mike Beedle
  - Scrum patterns @PLOPD4
- Ken Schwaber e Mike Cohn
  - 2002: Scrum Alliance



# Scrum usato da:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- Facebook
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Amazon
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

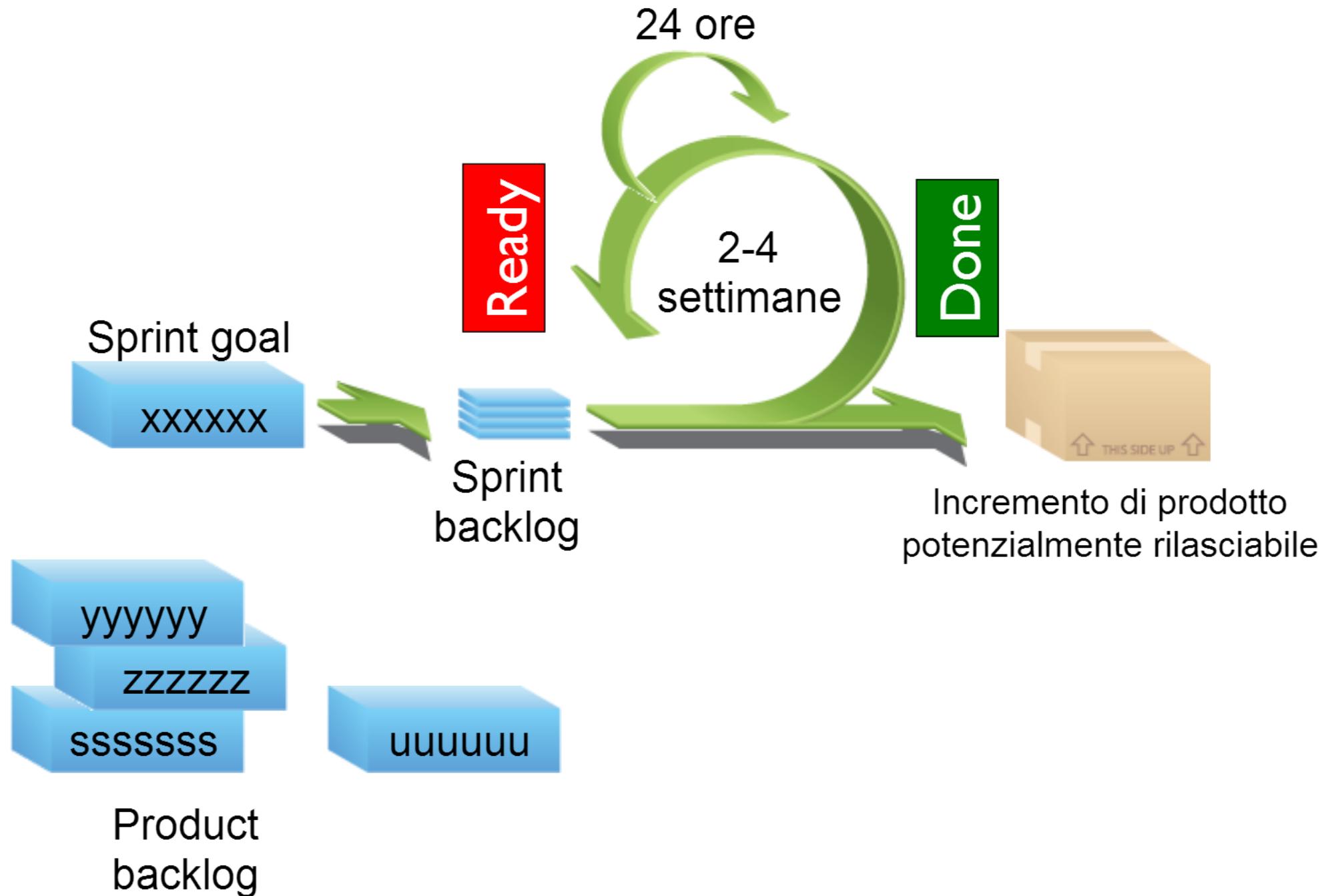
# Scrum usato per:

- Software commerciale
- Sviluppo in-house
- Sviluppi a contratto
- Progetti a prezzo prefissato
- Applicazioni finanza
- Applicazioni certificate ISO 9001
- Sistemi embedded
- the Joint Strike Fighter
- Applicazioni 24/7 con running time 99.999%
- Video giochi
- Sistemi life-critical
- Software di controllo satelliti
- Siti web
- Facebook, Amazon, Google
- Handheld software
- Sw per cellulari
- Network switching
- Applicazioni ISV
- Grandi sistemi

# Punti chiave di Scrum

- Metodo agile, parzialmente pianificato
- Sviluppo agile guidato da storie e test
- Team di sviluppatori auto organizzante
- Il prodotto cresce in sprint di durata fissa
- I requisiti sono catalogati nel product backlog
- Per ogni sprint, ogni persona del team sceglie i requisiti da realizzare da uno “sprint backlog”
- Meetings: reviews e “retrospettive”

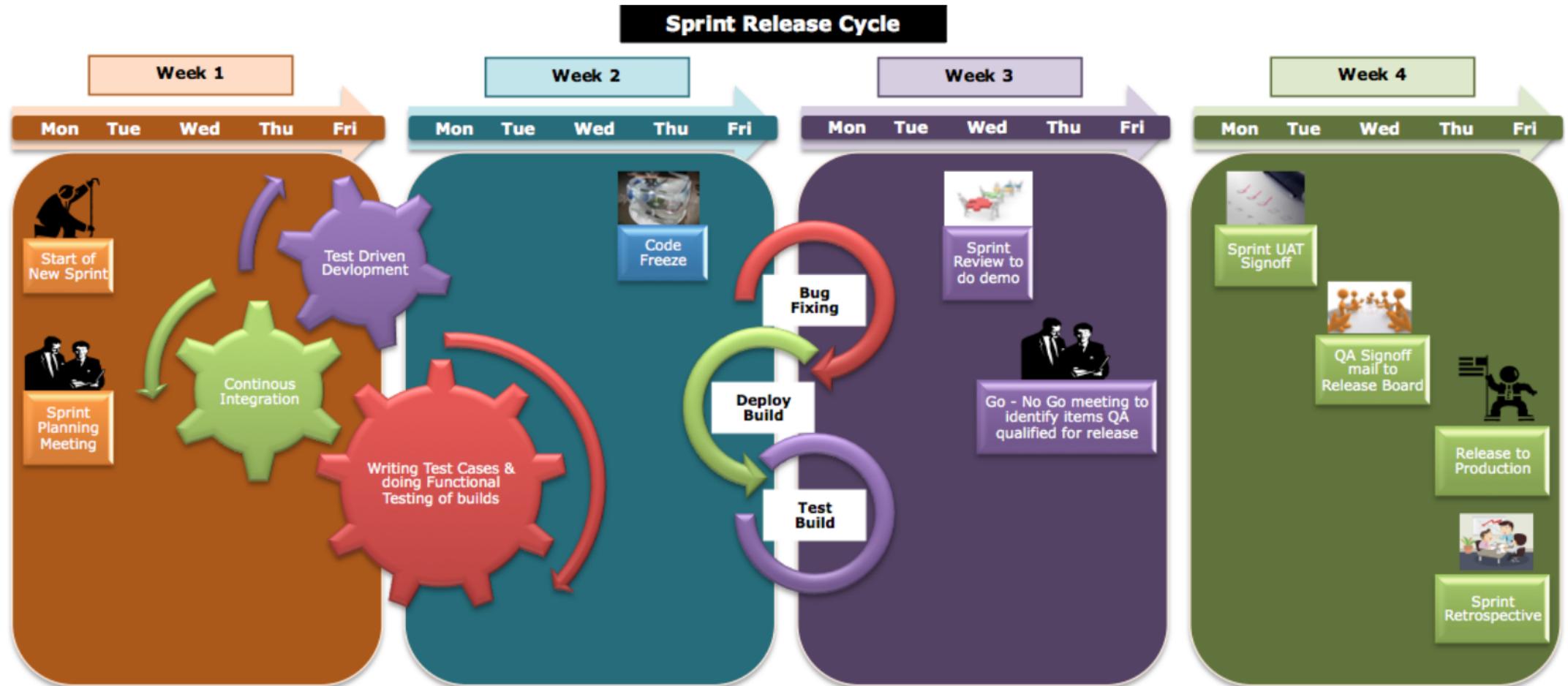
# Un processo Scrum si compone di **sprint**



# L'iterazione principale: lo sprint

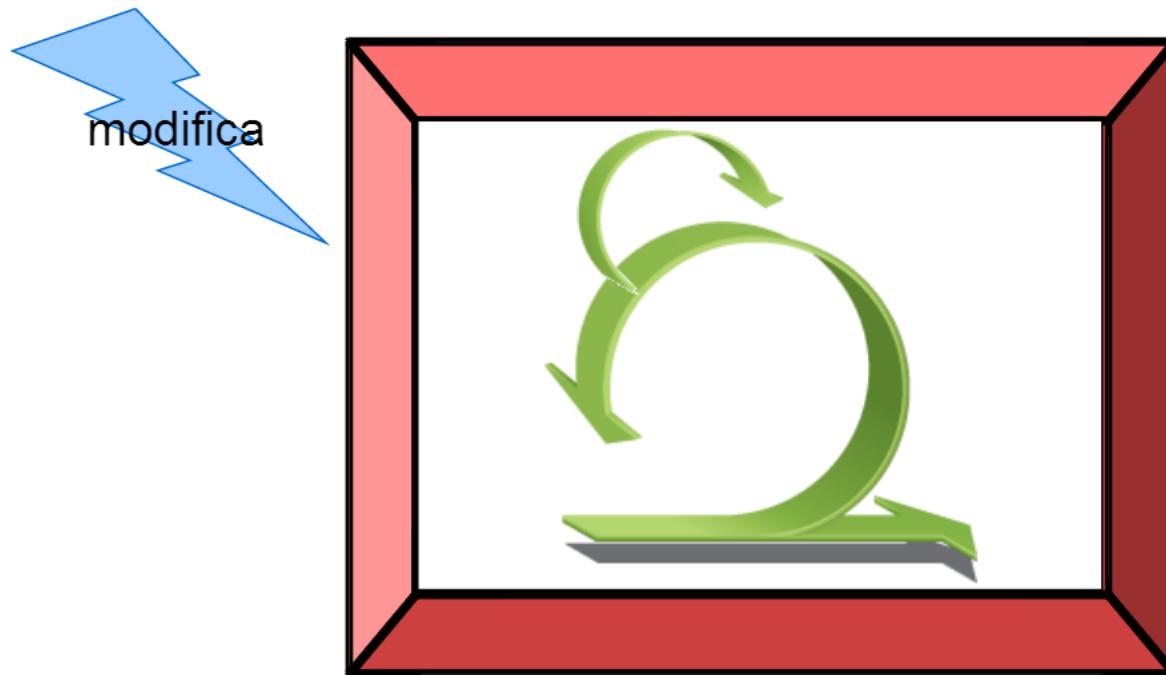
- Un processo Scrum è una serie di sprint (analoghi delle iterazioni RUP)
- Durata 2-4 settimane o un mese max
- Durata costante (migliora il ritmo)
- Ogni sprint include design codifica e test
- Ogni sprint estrae funzioni “Ready” dal product backlog e aggiunge codice “Done” al prodotto da mostrare al cliente

# Ciclo delle release in Scrum



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

# Niente cambiamenti al team durante lo sprint



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

# Scrum: ruoli rituali artefatti

## Ruoli

- Product owner
- ScrumMaster
- Team

## Rituali

- 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

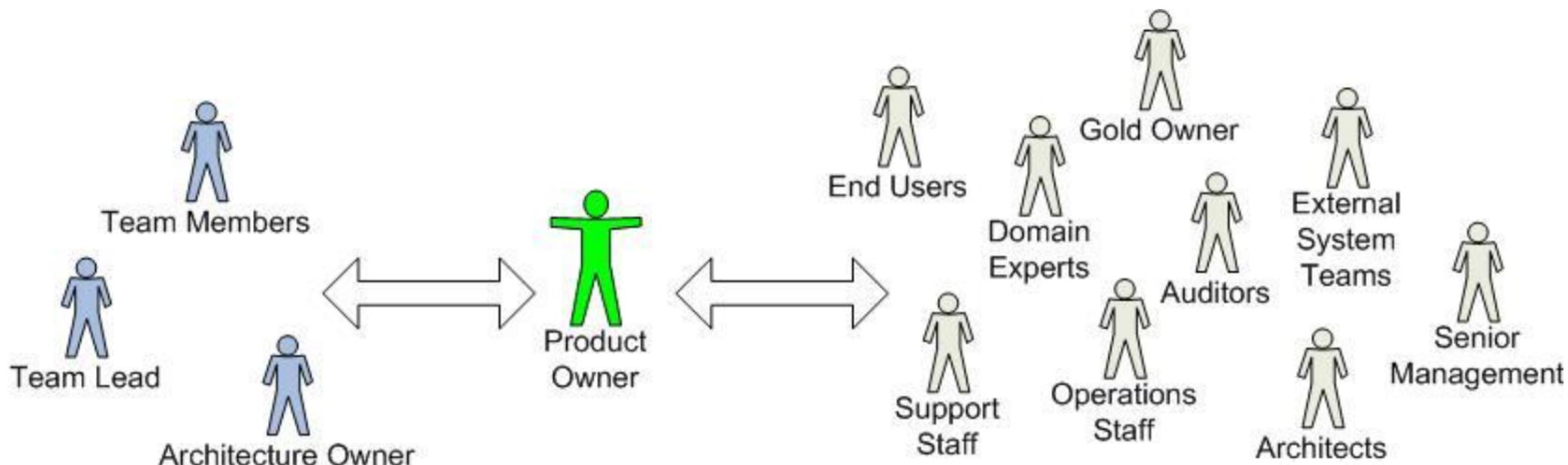
## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

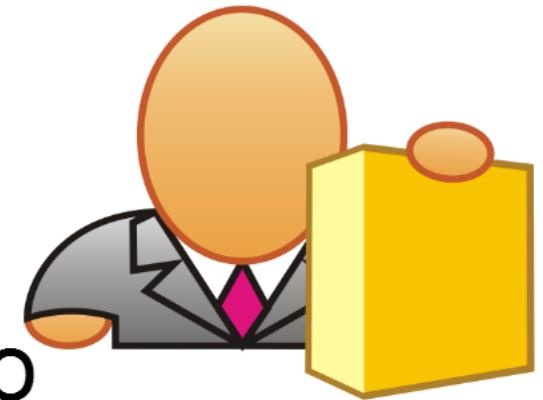
# Team Scrum

- Il Team Scrum (core) è formato da:
  - **Product Owner:** rappresenta gli stakeholder (la voce del cliente), scrive il *product backlog*, *user stories*,
  - **Development Team:** 3-9 membri con skill diversi responsabili della consegna di un PSI (Potentially Shippable Increment)
  - **Scrum Master:** facilita la corretta esecuzione del processo, elimina gli ostacoli
    - Meglio se non è coperto dalla persona con ruolo Product Owner
    - Non ha responsabilità di gestione del personale o di project management “tradizionale”

# Product owner



# Product owner



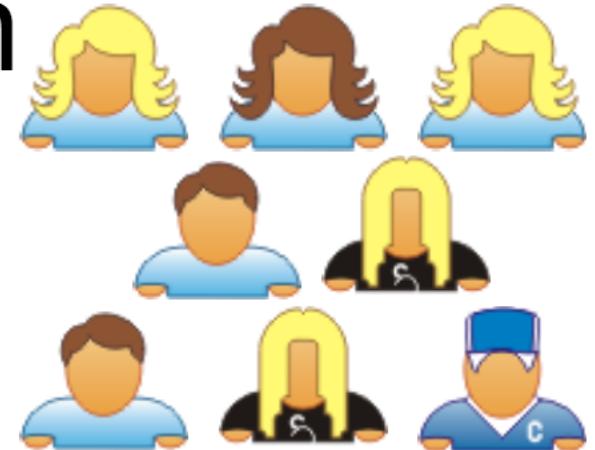
- Definisce le feature del prodotto
- Decide i rilasci: date e contenuti
- Responsabile del valore del prodotto (ROI)
- Mette in priorità le feature rispetto al loro valore di mercato
- Per ogni iterazione rivede lista delle feature e loro priorità ove necessario
- Accetta o rifiuta i risultati

# ScrumMaster



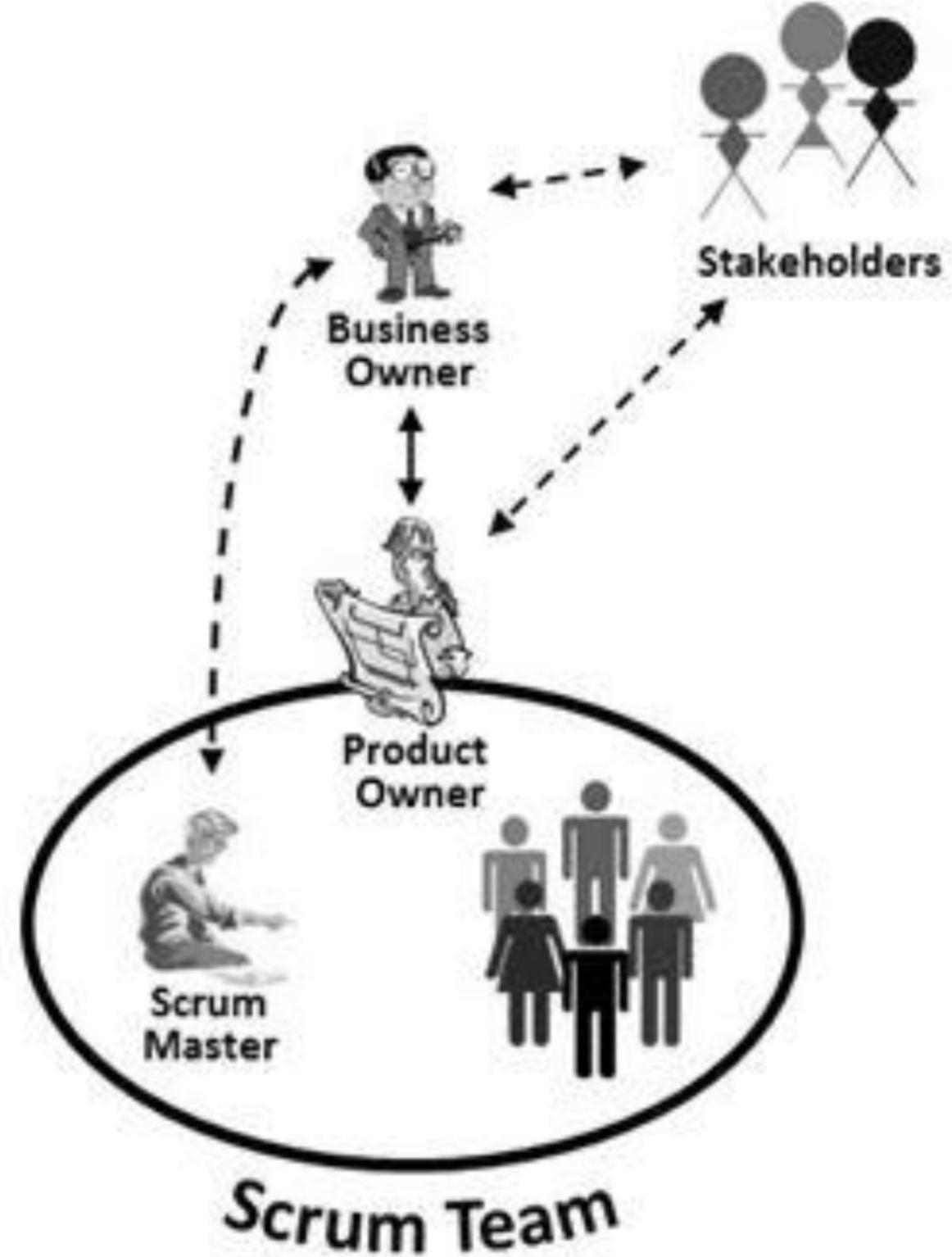
- Rappresenta il management
- Responsabile dei valori e pratiche Scrum
- Rimuove gli ostacoli
- Assicura il benessere del team
- Supporta la cooperazione di ruoli e funzioni
- Protegge il team da interferenze esterne

# Gli altri membri del team



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

# Scrum Team



# Il team



- Team autoorganizzante
- Modifiche al team solo in sprint diversi
- E' stabile per tutto lo sprint
- Tutti presenti nello stesso spazio di lavoro

# Scrum: i rituali

## Ruoli

- Product owner
- ScrumMaster
- Team

## Rituali

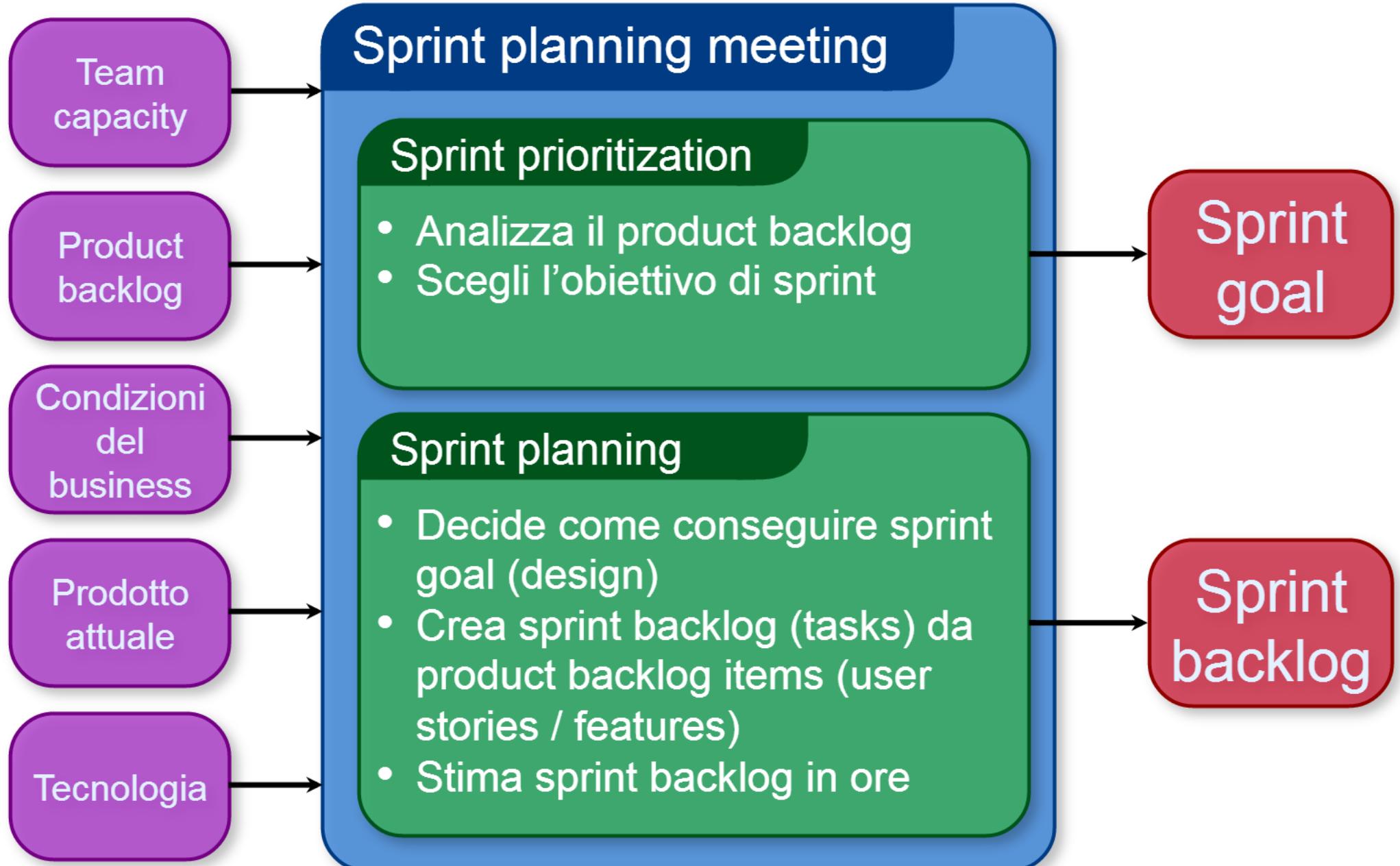
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

# I rituali Scrum

- **Sprint planning meeting:** cosa fare (sprint backlog) e come aggiornare il product backlog
  - 8 ore divise in due blocchi da 4
- **Daily scrum o stand-up:** ogni sviluppatore dice cosa ha fatto, cosa pianifica per oggi, che impedimenti ha trovato
  - 15 minuti, in piedi
- **Sprint Review:** riguarda il prodotto: cosa è stato o non è stato completato in questo sprint, demo
  - 4 ore al massimo
- **Sprint Retrospective:** riguarda il processo: cosa è andato bene e quali impedimenti sono stati trovati
  - 3 ore al massimo



# Pianificazione dello sprint

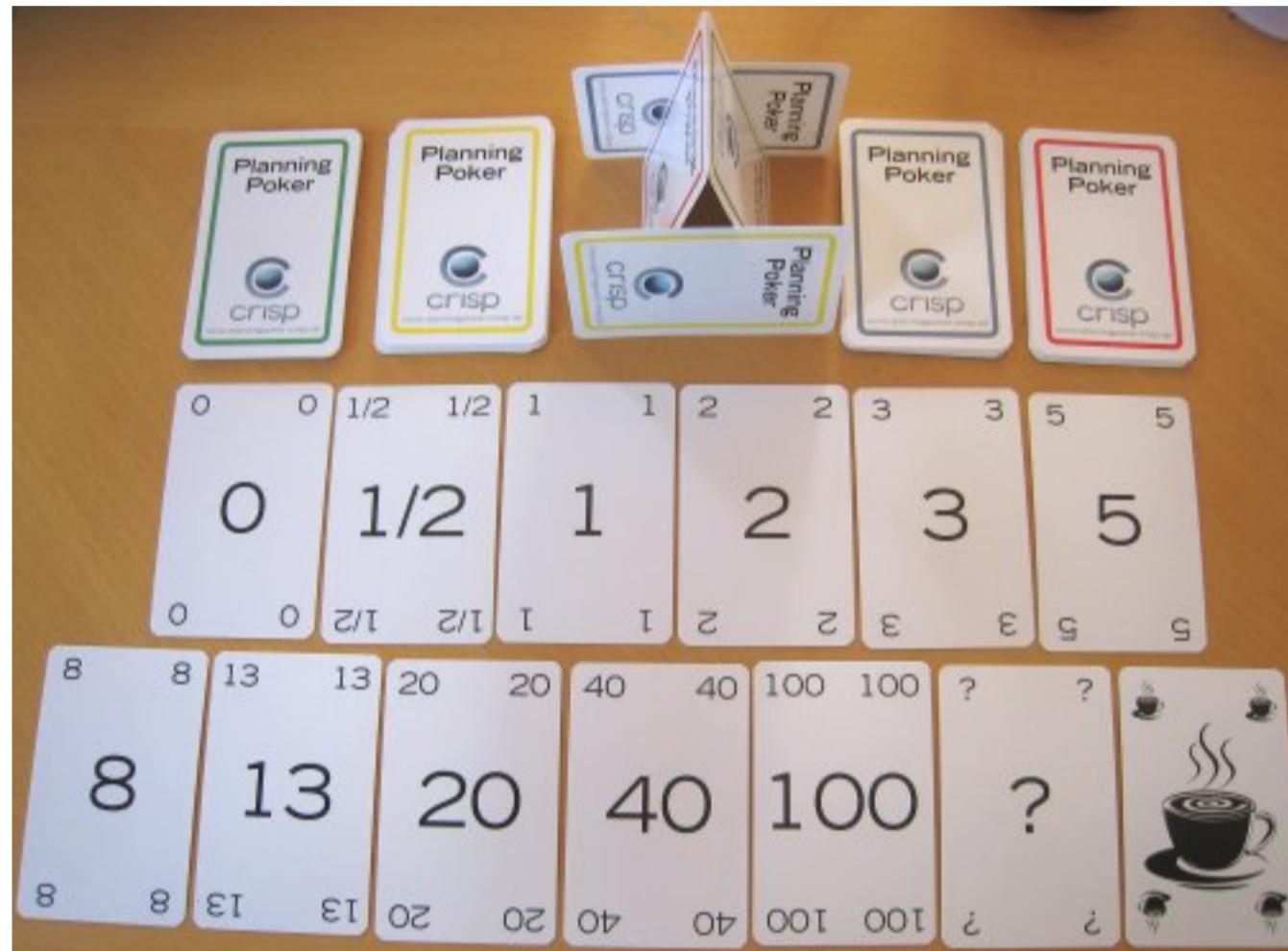
- Il team sceglie dal product backlog gli elementi che verranno certamente realizzati nello sprint
- Studio dell'architettura di alto livello
- Si crea lo sprint backlog: i compiti da fare
  - Identificazione e stima di ciascun compito (1-16 hours)
  - Uso di Planning Poker

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



# Scrum quotidiano

- Modalità
  - quotidiano
  - 15-minuti
  - In piedi
- Niente problem solving
  - Chiunque può assistere
  - Possono parlare solo i pig: membri del team, ScrumMaster, Product Owner
  - Scopo: evitare troppi incontri inutili



# Tre domande a tutti

1

Che hai fatto ieri?

2

Che farai oggi?

3

Prevedi problemi?

- Non servono allo ScrumMaster per vedere il progresso
- Le risposte sono promesse ai compagni di team

# La revisione dello sprint (sprint review)

- Il team presenta ciò che ha ottenuto con lo sprint rispetto al prodotto: come questo è “cresciuto”
- E’ una demo delle nuove funzioni o della nuova architettura
- Serve per verificare il product backlog
- Informale
  - 2 h di preparazione
  - Niente slide
- Partecipa tutto il team
- Invitare tutti



# Sprint retrospective

- Riguarda il processo: come sta andando?
- Occorre analizzare periodicamente cosa va e cosa non va in ciascuno sprint
- Quando: dopo ogni sprint, dopo la review
- Partecipano tutti:
  - ScrumMaster
  - Product owner
  - Team
  - Forse clienti ed utenti ed altri stakeholders

# Start / Stop / Continue

Il team discute cosa vorrebbe fare:

Start doing

Stop doing

Ci sono vari  
modi alternativi  
di definire una  
retrospettiva

Continue doing

# Lo scopo della retrospettiva

- Esaminare come è andato l'ultimo Sprint riguardo a persone, relazioni, processi e strumenti;
- 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 raramente ha senso cancellare uno sprint

# Scrum: gli artefatti

## Ruoli

- Product owner
- ScrumMaster
- Team

## Rituali

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

## Artefatti

- Product backlog
- Sprint backlog
- Burndown charts

# Product backlog

- Lista dei requisiti
- Definita in modo tale che ciascun elemento abbia valore per gli utenti o i clienti del prodotto
- Messa in priorità da product owner
- Priorità ridefinite all'inizio di ogni sprint



Questo è il  
product 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        |

# 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.

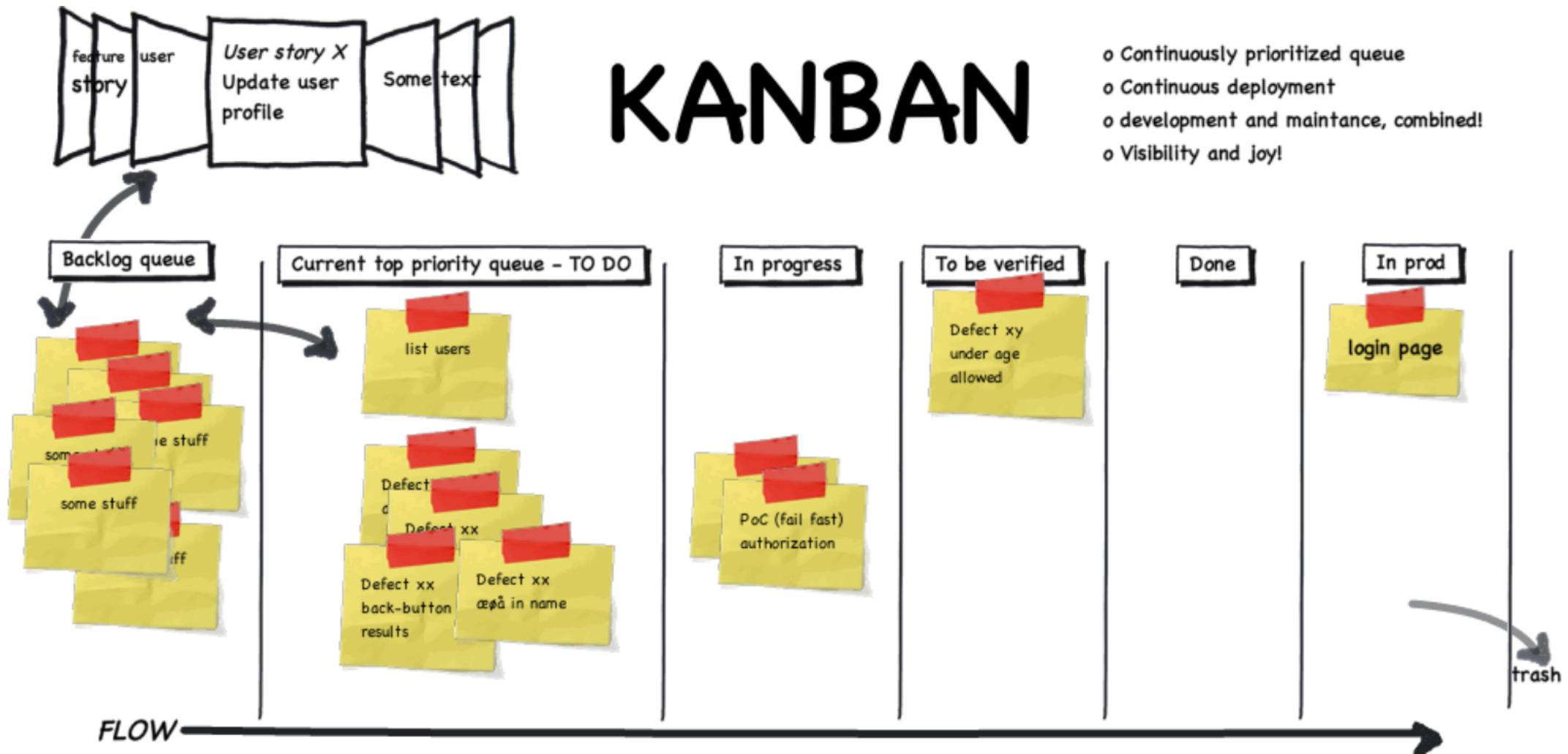
# Gestione dello sprint backlog

- 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

# Gestione dello sprint backlog

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

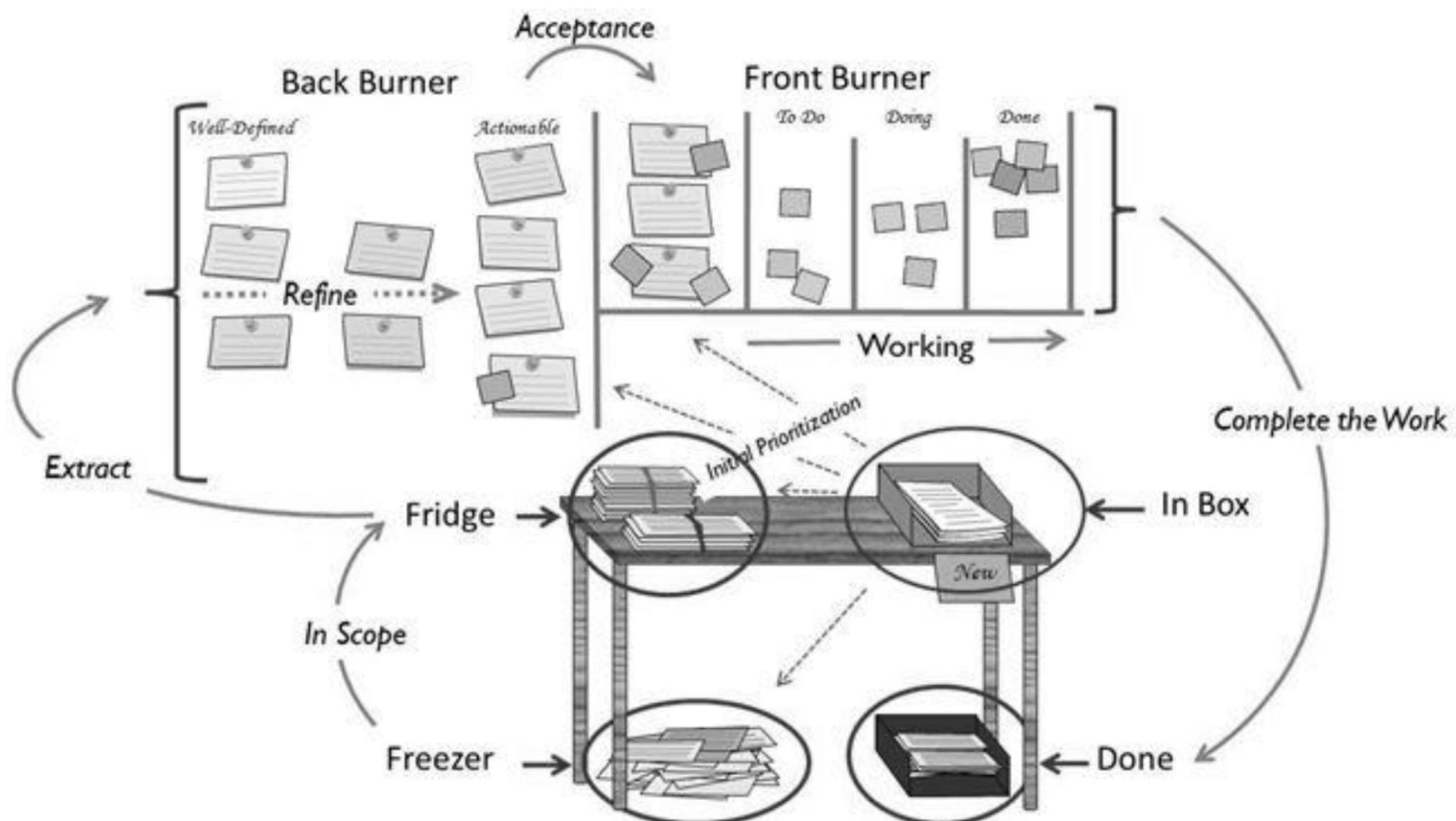
# KANBAN



Created by Ole Morten Amundsen using Mockito. Use as you please.

- o Continuously prioritized queue
- o Continuous deployment
- o development and maintenance, combined!
- o Visibility and joy!

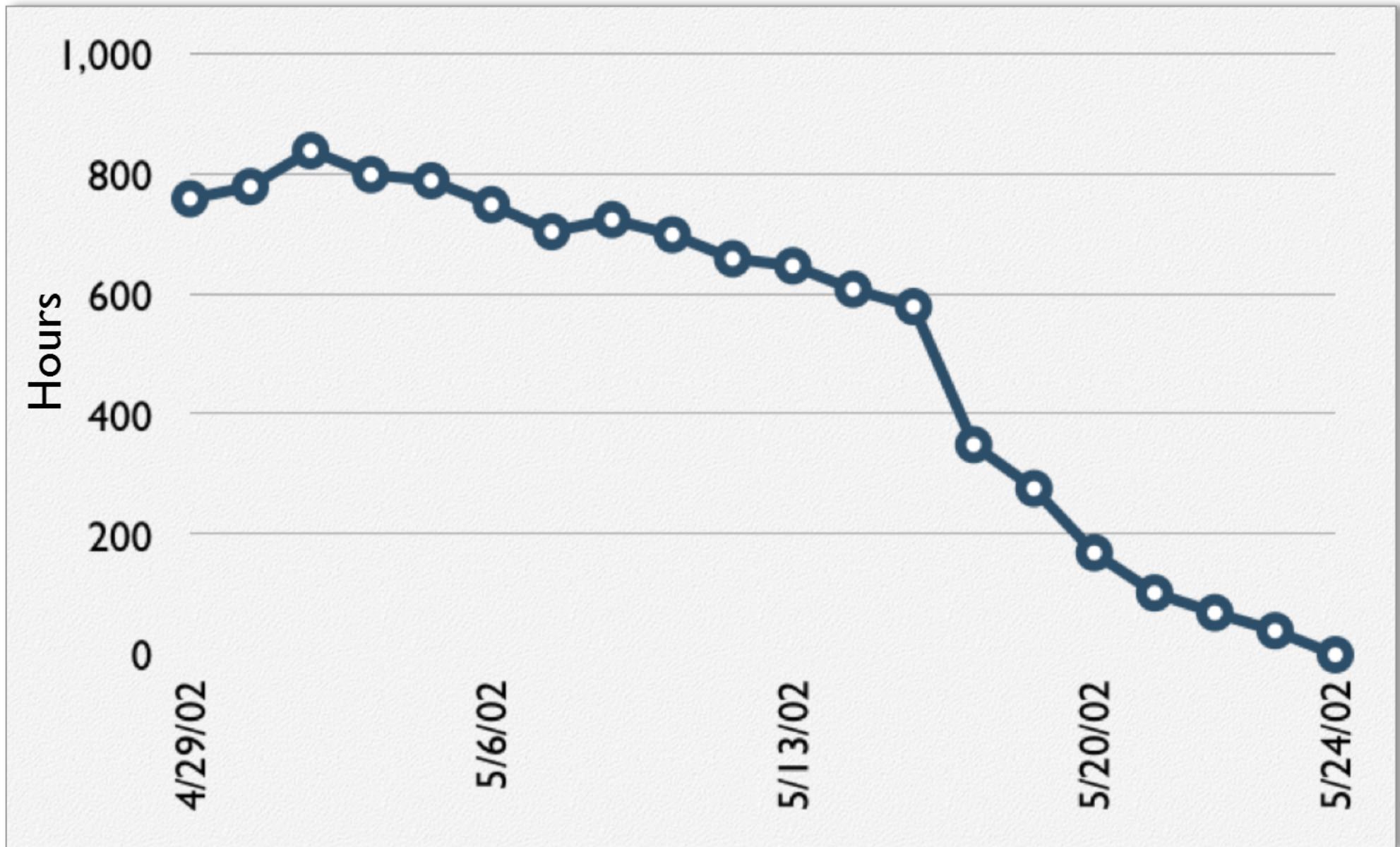
# Il ciclo dei compiti in uno sprint



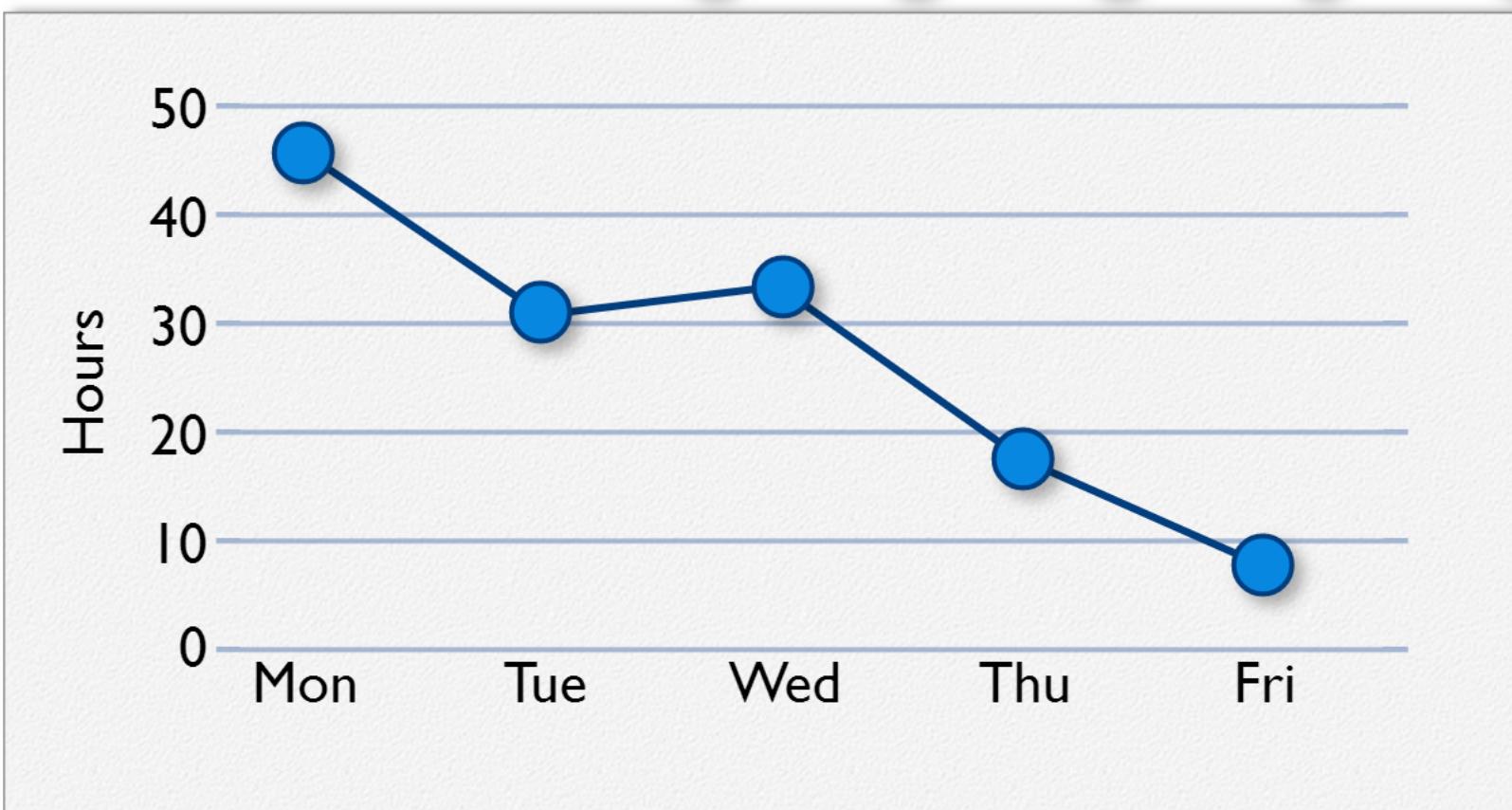
# Uno sprint backlog

| 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

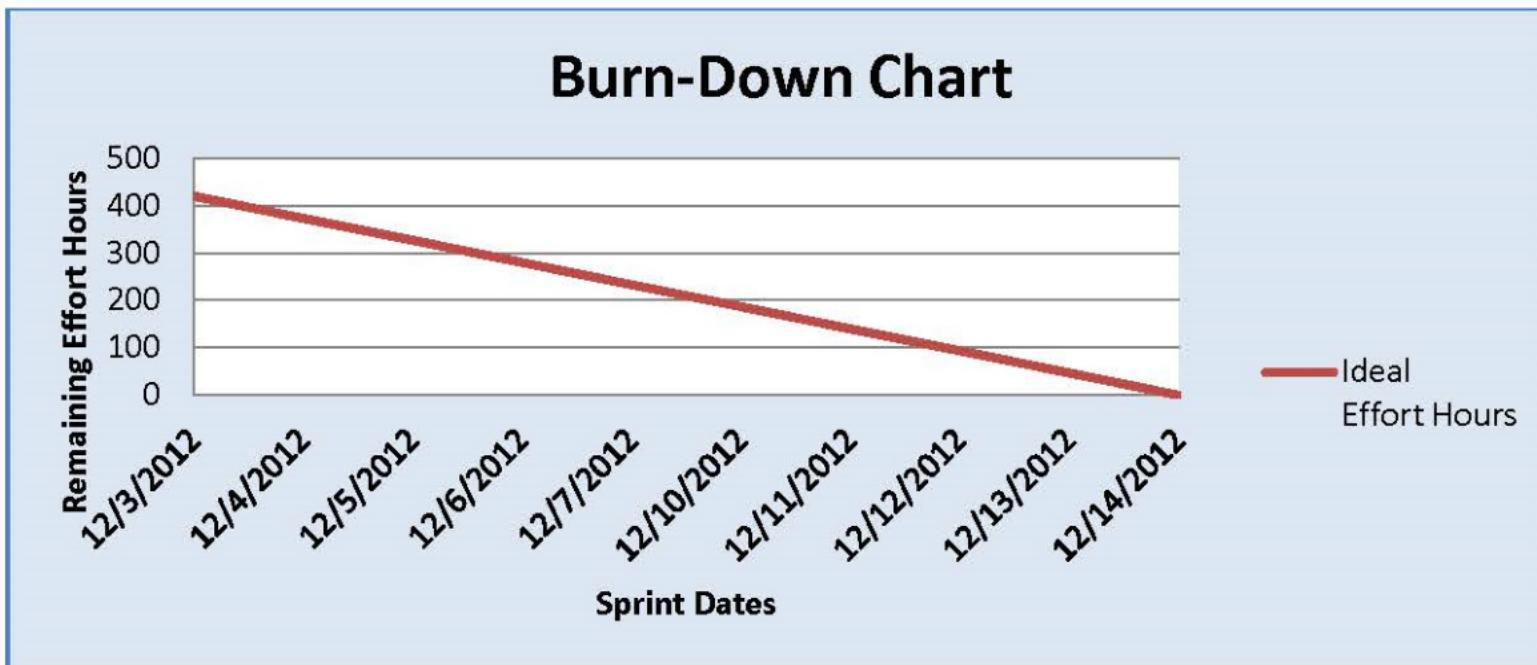


| Tasks                   | Mon | Tues | Wed | Thur | Fri |
|-------------------------|-----|------|-----|------|-----|
| Code the user interface | 8   | 4    | 8   |      |     |
| Code the middle tier    | 16  | 12   | 10  | 7    |     |
| Test the middle tier    | 8   | 16   | 16  | 11   | 8   |
|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



# Esempio

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



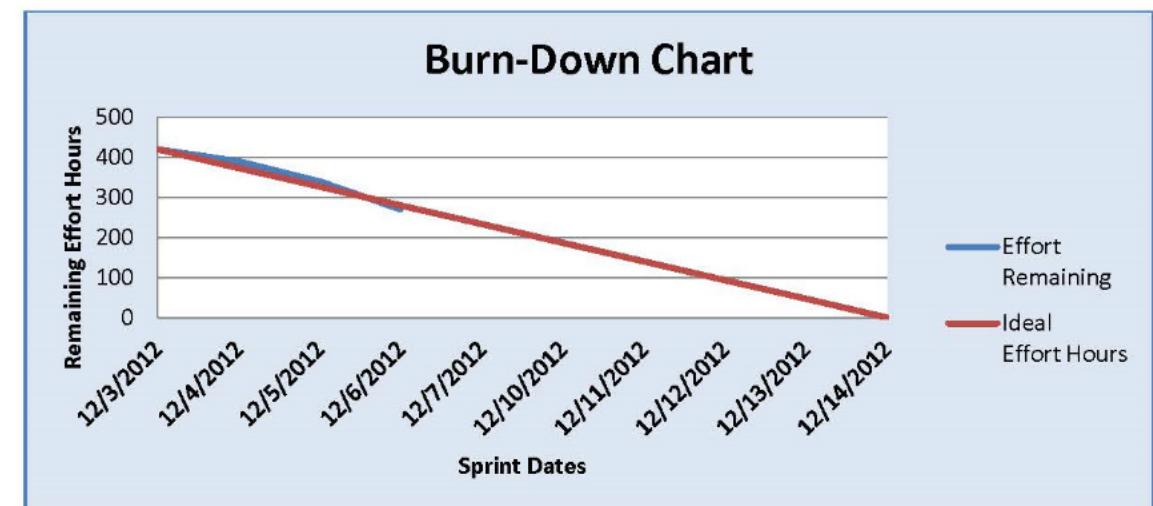
# Stima continua dello sforzo rimanente

Ogni partecipante sceglie un compito e stima il tempo rimanente.

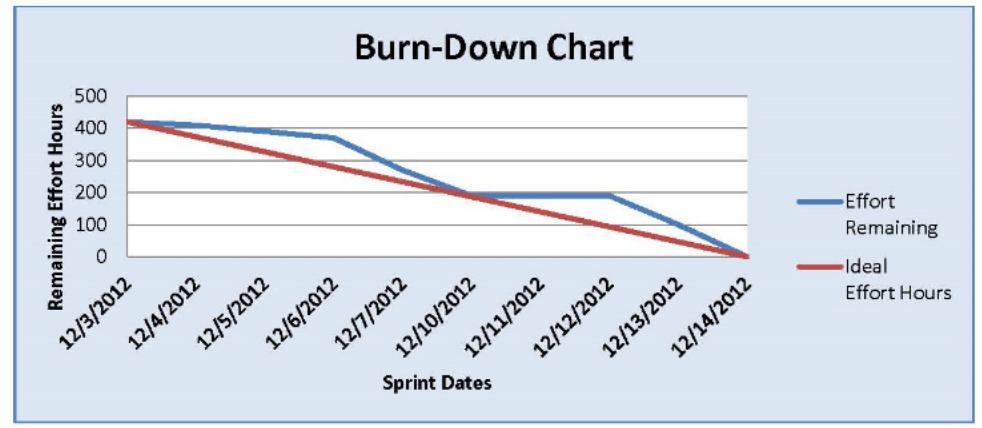
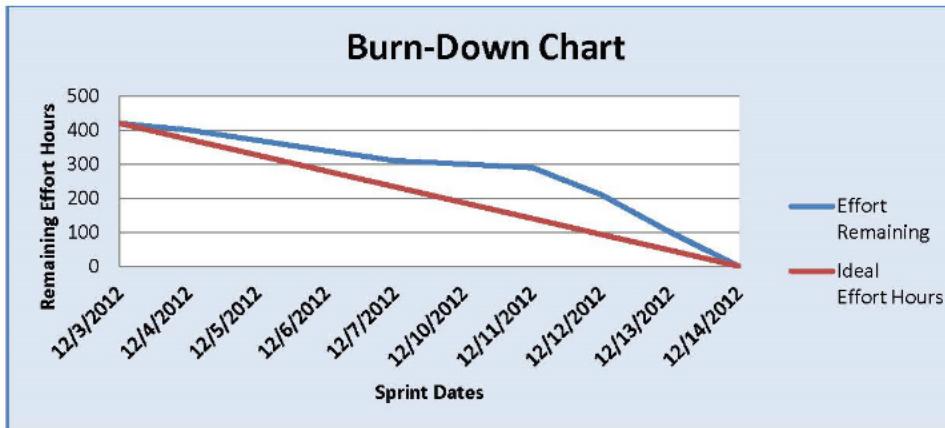
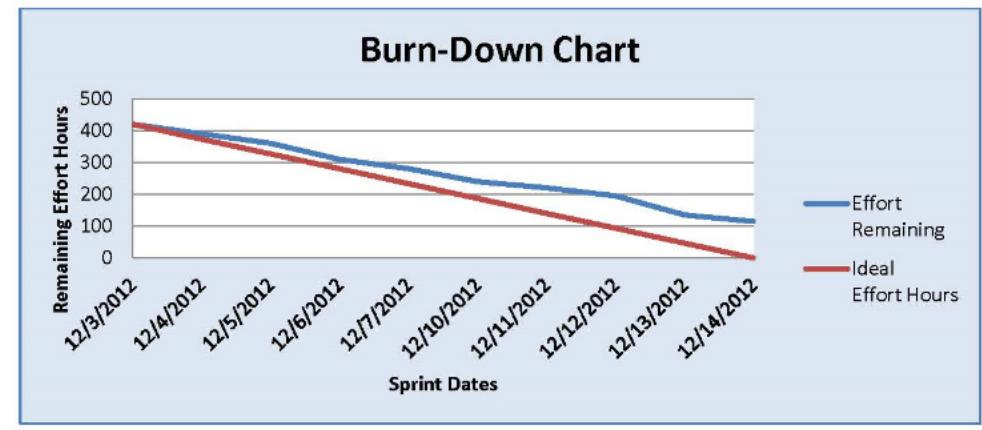
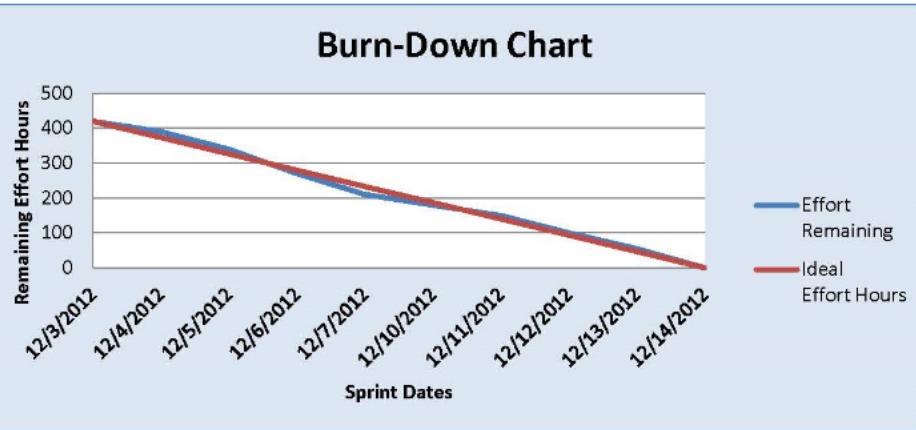
Per es. dopo il primo giorno sono state spese 6 h sul compito 1. Il programmatore stima che rimangano altre 6 ore (quindi 2h oltre la stima di 10h).

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                          |



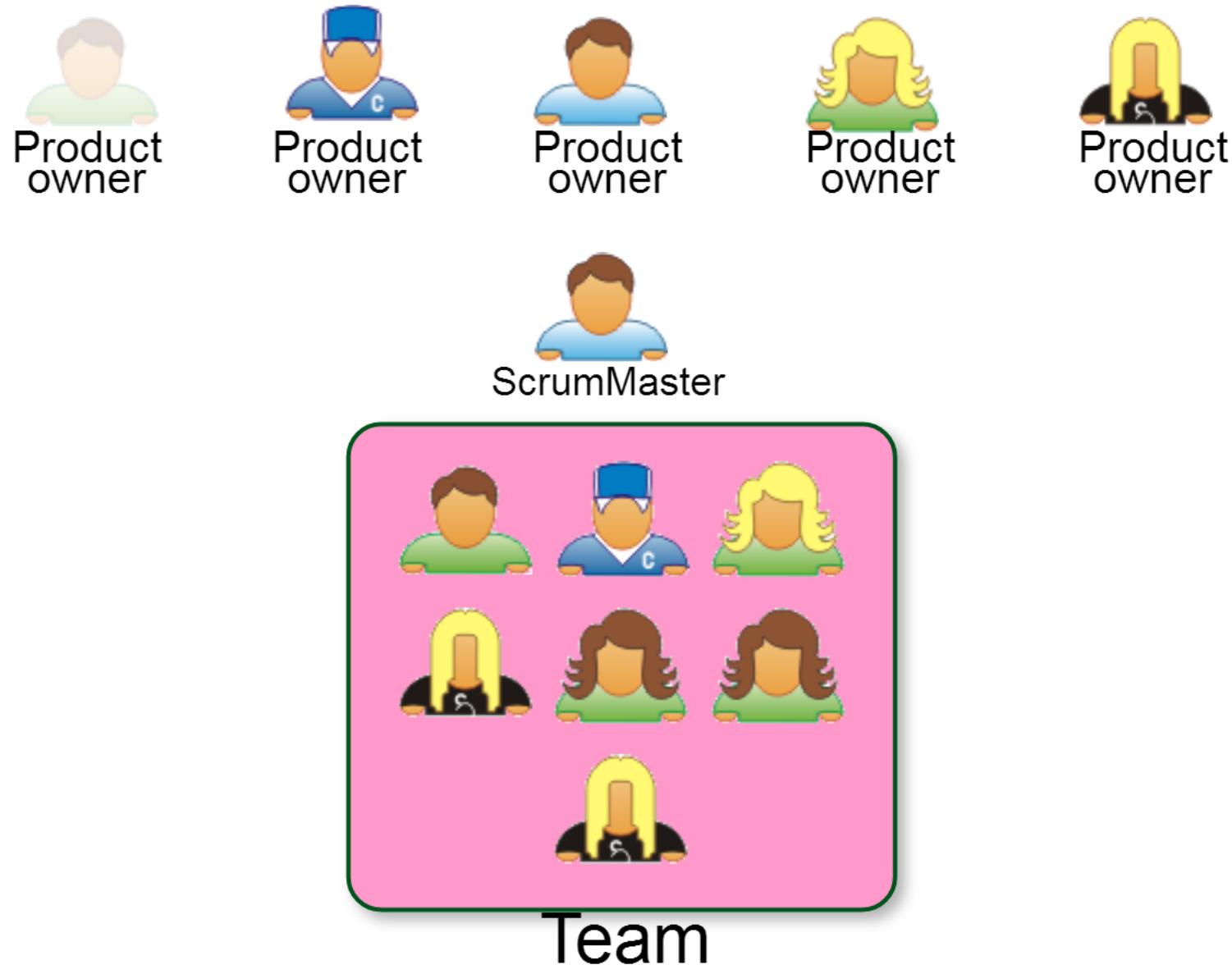
# Varie situazioni



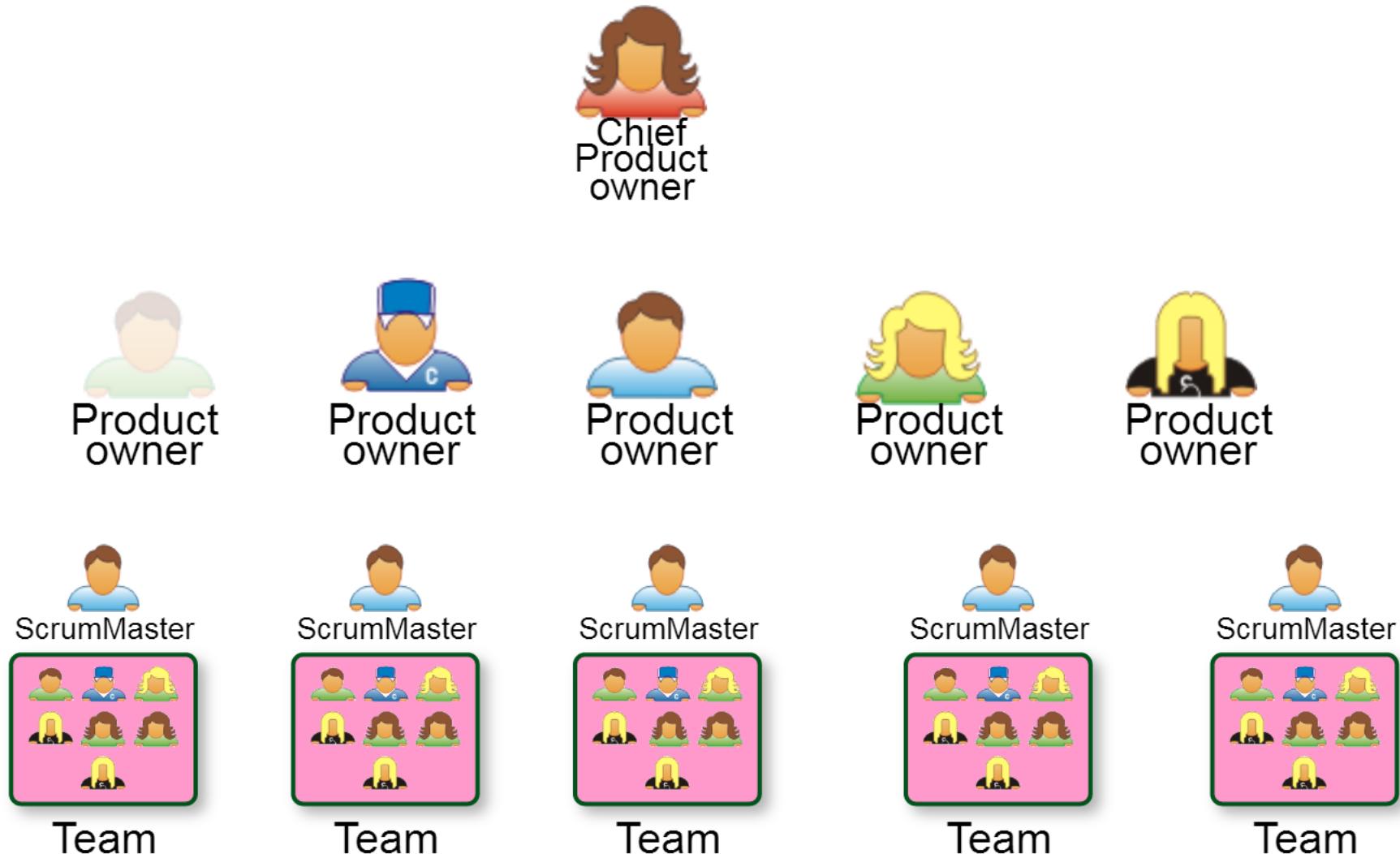
# Scalabilità

- Team:  $7 \pm 2$  persone
  - Si scala su grossi sistemi con scrum di scrum
- fattori
  - Tipo dell'applicazione
  - Dimensione dei team
  - Dispersione dei team
  - Durata del progetto
- Alcuni progetti Scrum hanno coinvolto oltre 500 persone

# Team unico, molteplici product owner



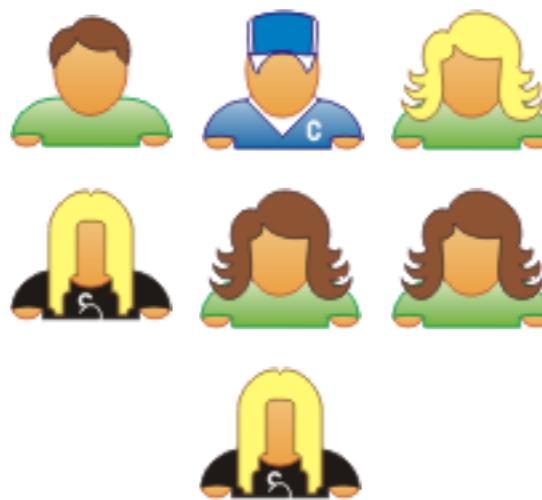
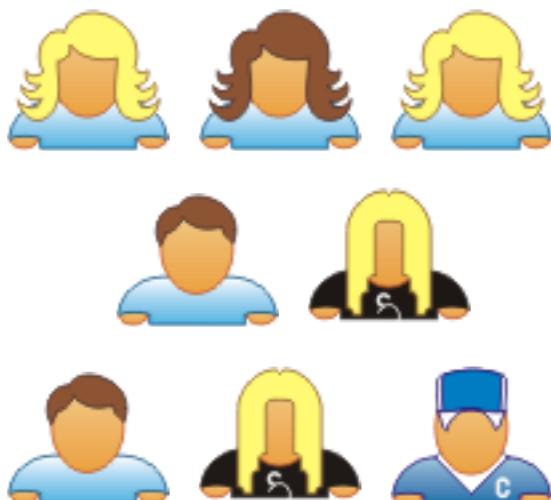
# Molteplici team, prodotto unico



# Scrum di scrum



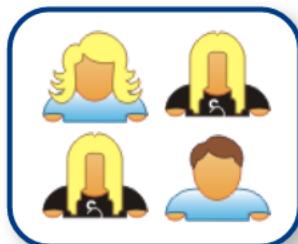
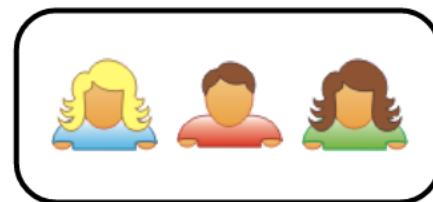
Metascrum  
degli ambasciatori



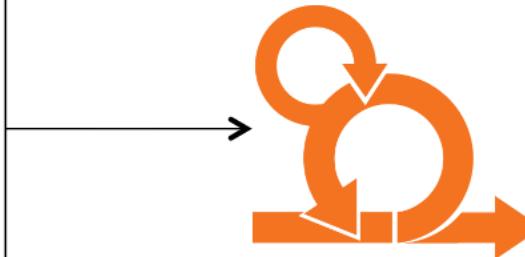
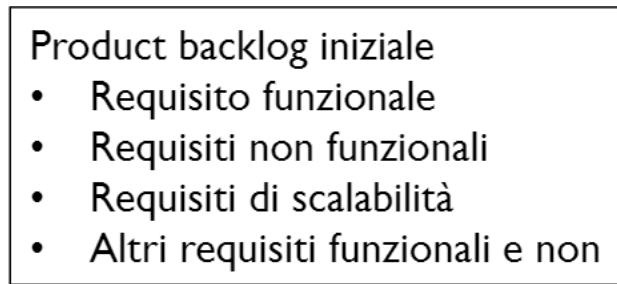
# J. Sutherland sullo scrum di scrum

- *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.*

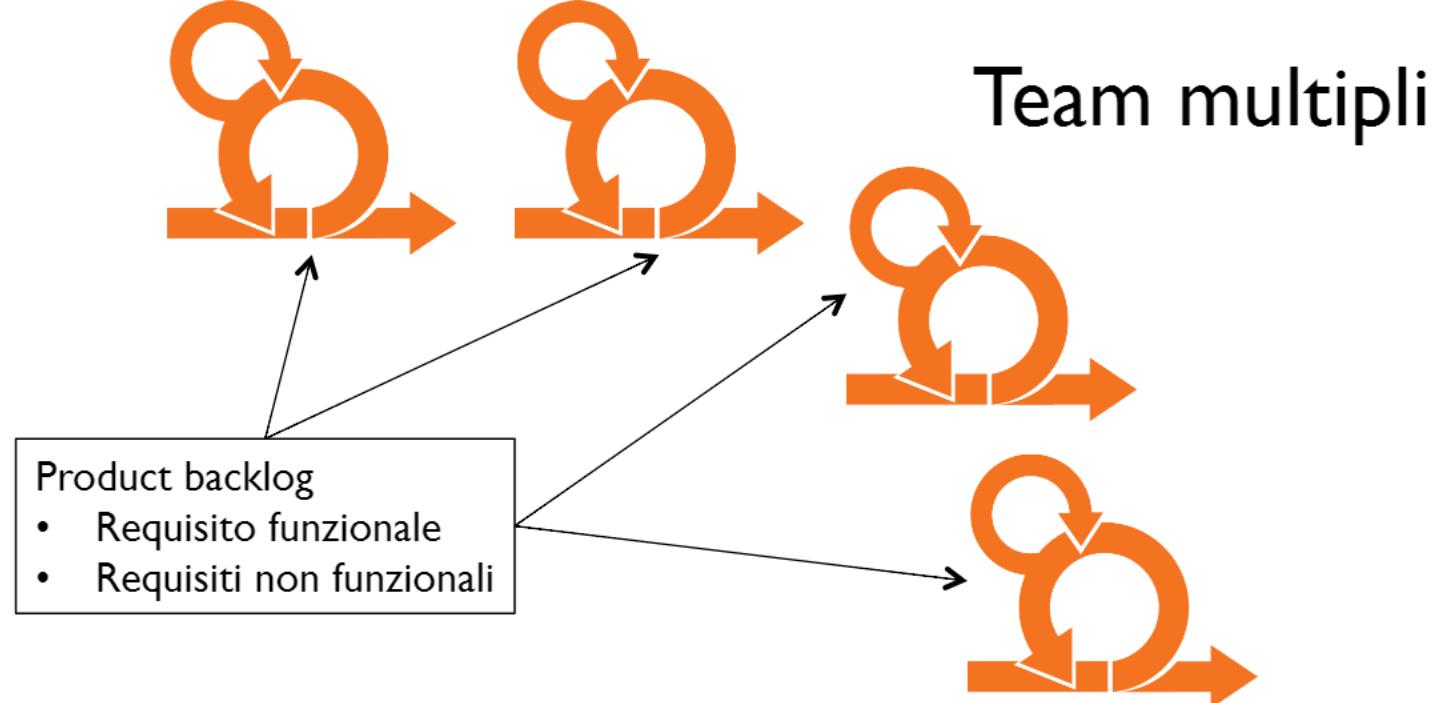
# Scrum di scrum di scrum



# Coordinare team multipli



Team unico



Team multipli

# Problemi tipici con Scrum

1. Ignoranza dei valori agili e di Scrum
2. Prodotto software non testato alla fine dello sprint  
(cattiva definizione di *Done*)
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

# Riferimenti utili

- [www.scrumalliance.org](http://www.scrumalliance.org)
- [www.controlchaos.com](http://www.controlchaos.com)
- [www.mountaingoatsoftware.com/scrum](http://www.mountaingoatsoftware.com/scrum)

# 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?

