

**Calogero 'Kalos' Bonasia**  
semplificazione e miglioramento  
dei processi aziendali  
[www.bonasia.info](http://www.bonasia.info)

***Seminario di Orientamento al Lavoro***

***Palermo 25/10/2017***

***Università di Palermo***

***Dipartimento di Matematica ed Informatica***



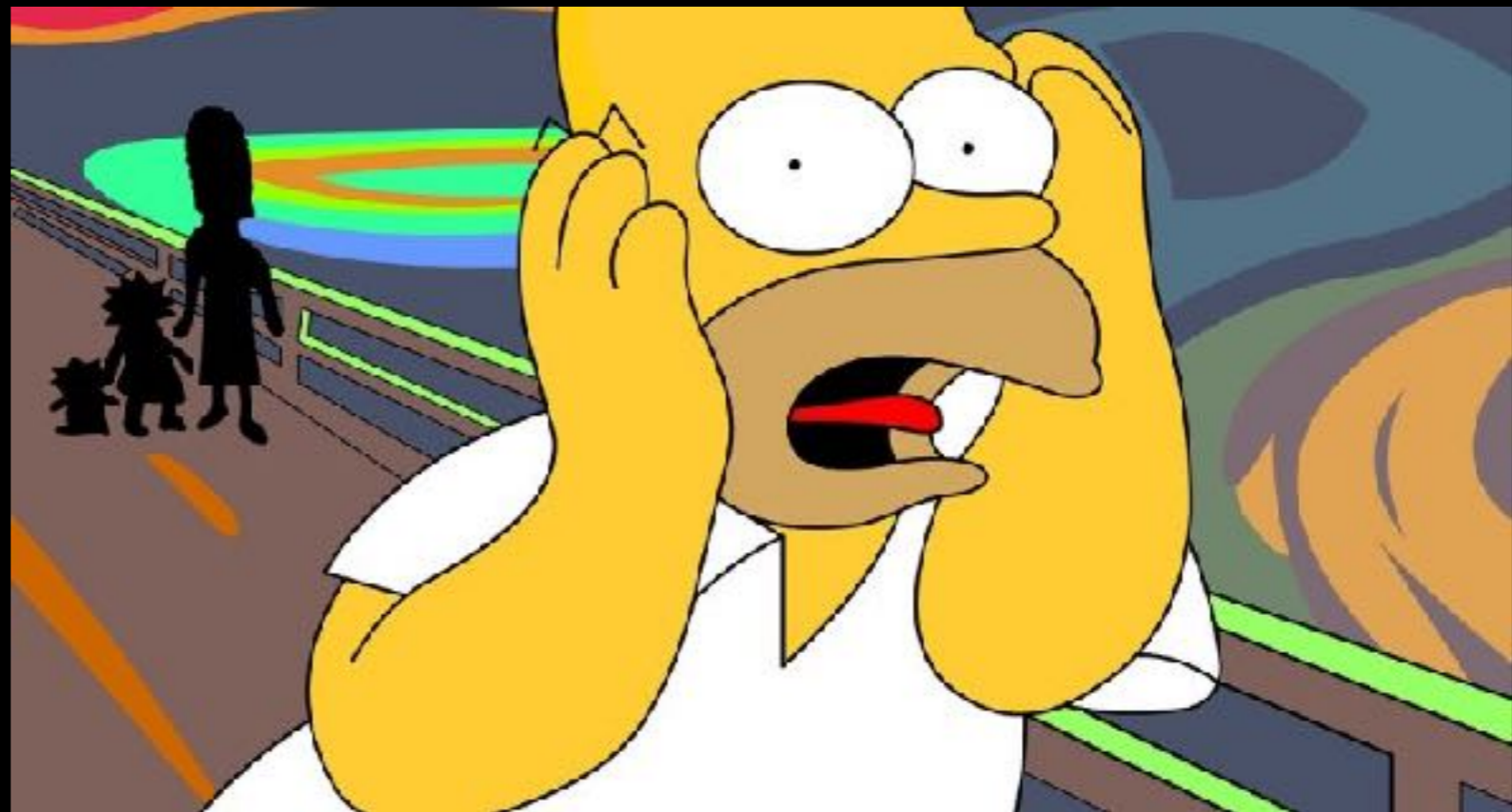
UNIVERSITÀ  
DEGLI STUDI  
DI PALERMO

## perché ?

Quando inizio un nuovo workshop o una nuova attività di coaching, io pongo spesso la domanda:

**sapete dirmi qual è il problema principale  
nel mondo del software ?**

domanda che scatena le risposte più disparate: dalla comprensione dei requisiti alla capacità di rispettare la pianificazione





# il mondo "reale"



How the customer explained it



How the project leader understood it



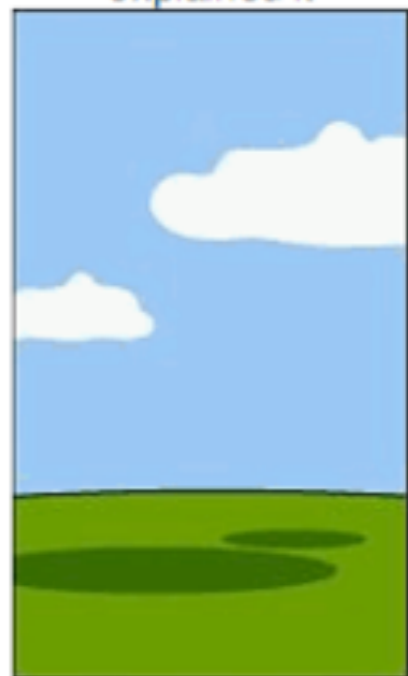
How the engineer designed it



How the programmer wrote it



How the sales executive described it



How the project was documented



What operations installed



How the customer was billed



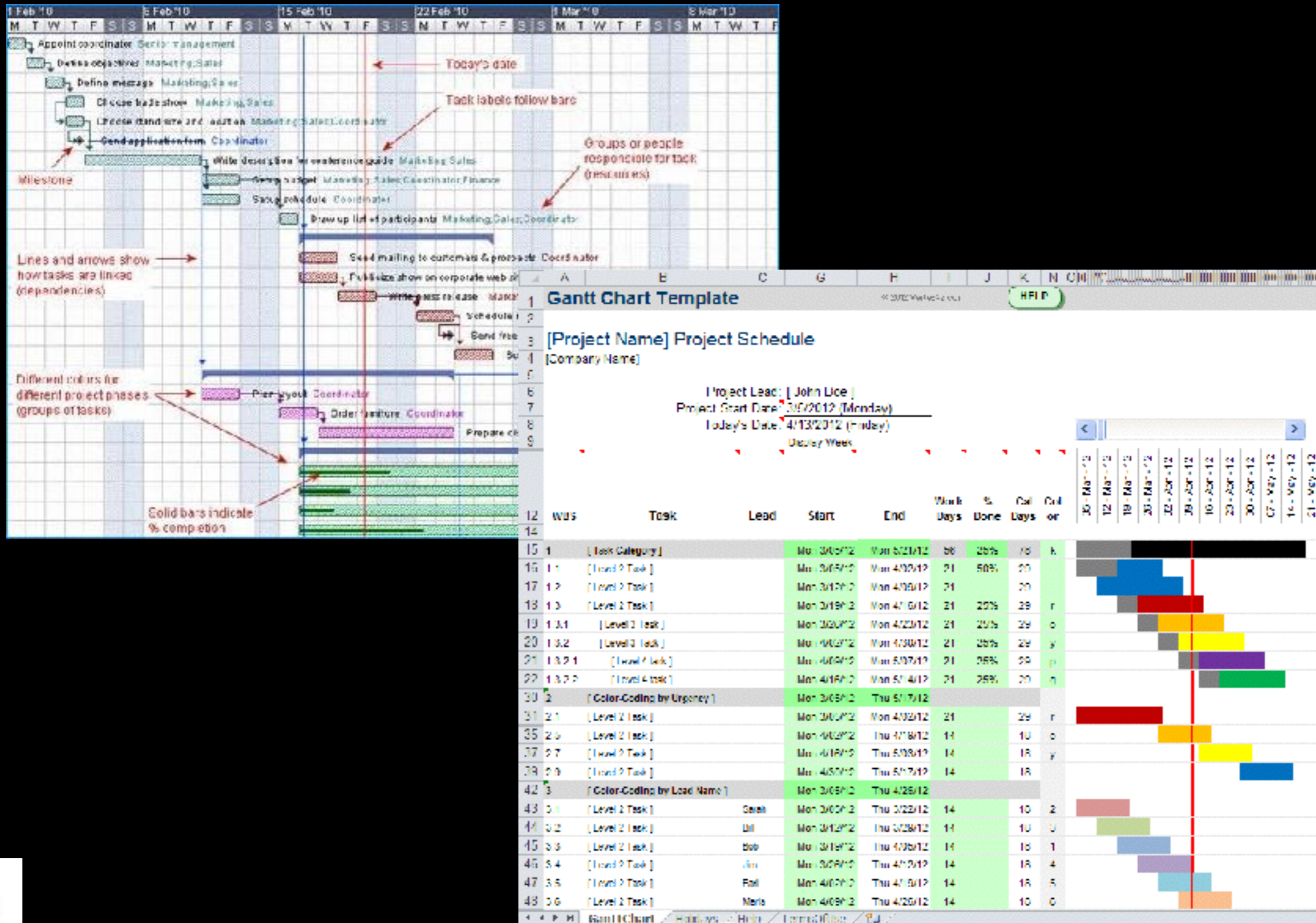
How the helpdesk supported it



What the customer really needed



# “Progettazione Management”





# lavorare con intelligenza emotiva

La verità è un'altra: il software è un artefatto complesso  
realizzato da un sistema adattivo complesso:  
**le persone, cioè noi**

Un gruppo di  
persone che devono  
imparare a  
collaborare ed essere  
una squadra  
non semplicemente  
stare sedute l'una a  
fianco all'altra



**Gli individui e le interazioni**  
*più che* **i processi e gli strumenti**

**Software funzionante**  
*più che* **documentazione esaustiva**

**Collaborazione con il cliente**  
*più che* **la negoziazione dei contratti**

**Rispondere al cambiamento**  
*più che* **seguire un piano**



1. le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano
2. fondiamo i **progetti** su **individui motivati**, dando loro l'ambiente e il supporto di cui hanno bisogno
3. i processi agili promuovono uno sviluppo sostenibile
4. tutti i soggetti coinvolti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante



- 1. il software funzionante è il principale metro di misura del progresso**
- 2. la massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua**
- 3. consegnare frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi**





- 1. committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto**
- 2. una conversazione faccia a faccia è il modo più efficace ed efficiente per comunicare con il team ed all'interno del team**



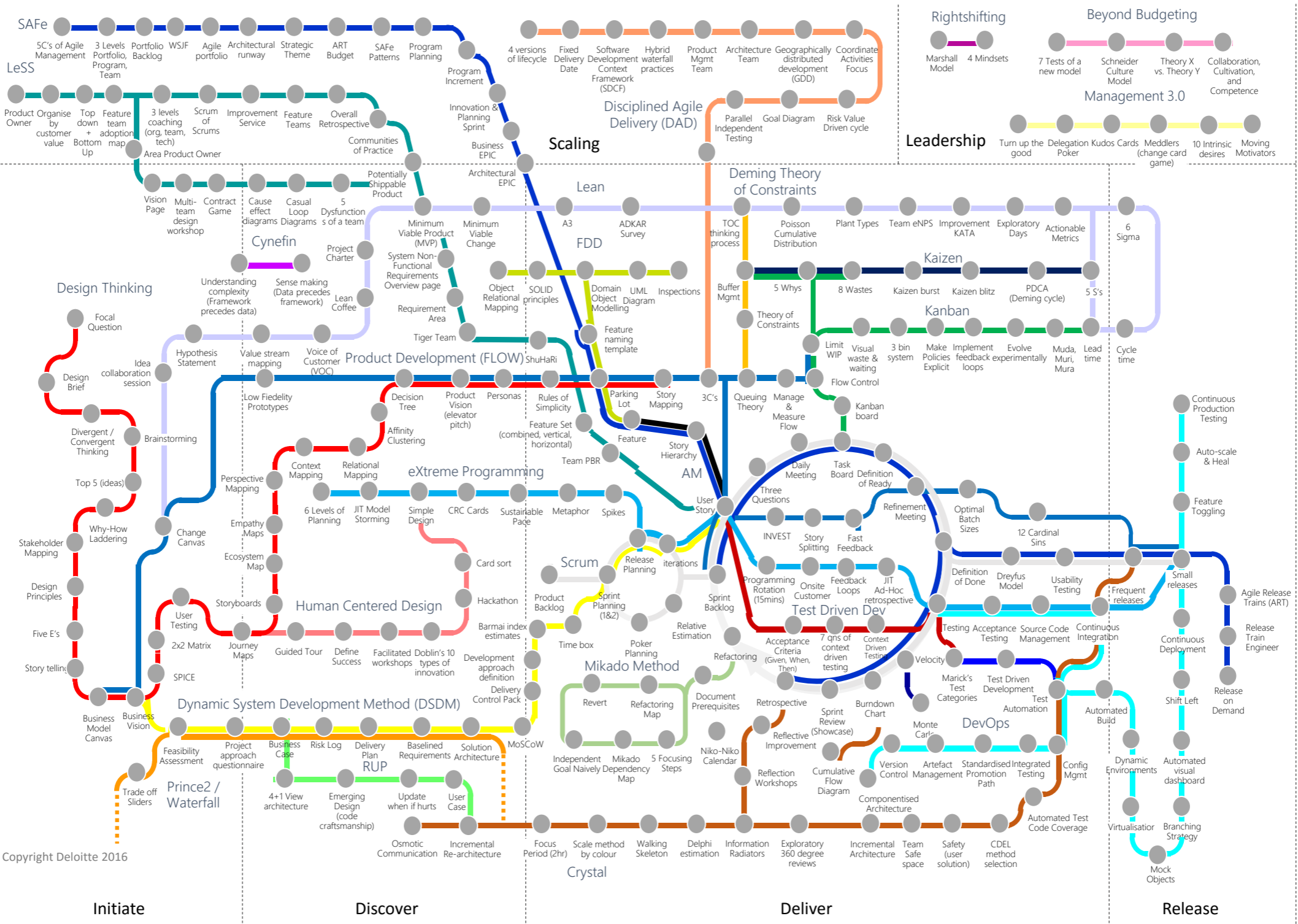
- 1. accogliamo i cambiamenti nei requisiti anche a stadi avanzati del processo di sviluppo**
- 2. i processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente**
- 3. ad intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza**



# Agile: landscape

Deloitte Agile Landscape v2

Developed by Christopher Webb. For more info please contact : [Agile@deloitte.com.au](mailto:Agile@deloitte.com.au)





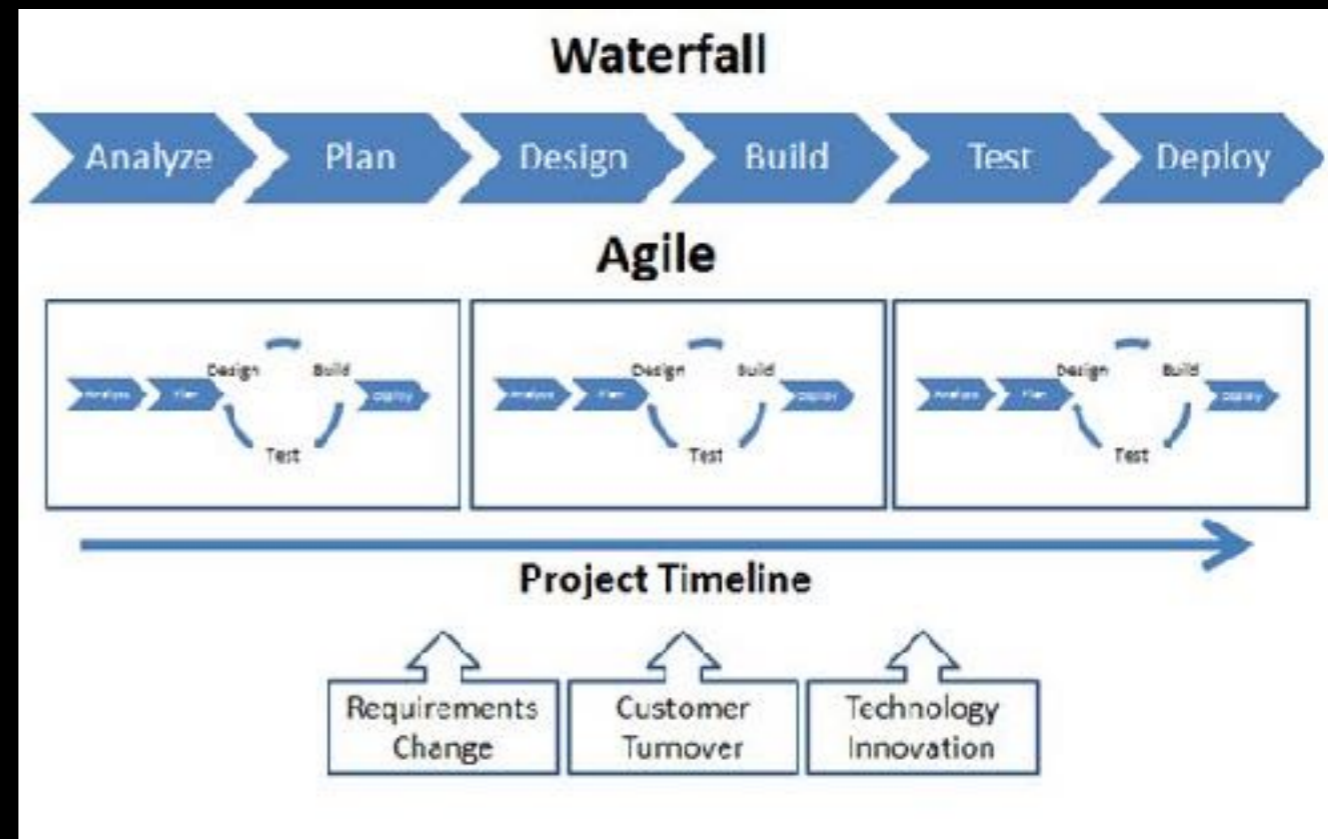
# Scrum: ??



Scrum è un framework agile per la gestione del ciclo di sviluppo del software, iterativo ed incrementale

concepito per gestire progetti e prodotti software o applicazioni di sviluppo, creato e sviluppato da

**Ken Schwaber**  
e  
**Jeff Sutherland**



**concentrazione**

**coraggio**

**trasparenza**

**impegno**

**rispetto**



**Product Owner (PO)**

**Scrum Master (SM)**

**Development Team  
(DevTeam)**





## Product Owner (PO)

- 1. aiuta a descrivere il prodotto (“è il cliente”)**
- 2. identifica il valore per l’utente finale**
- 3. pianifica la *roadmap* di prodotto ed è presente durante tutto lo *Sprint***
- 4. ordina le caratteristiche del prodotto in funzione del valore (collaborando con DevTeam e Stakeholders)**



# Scrum Master (SM)

- 1. supporta il DevTeam e il Product Owner durante tutto il progetto**
- 2. facilita le riunioni**
- 3. aiuta a rimuovere gli impedimenti per il DevTeam**
- 4. affianca le persone dello Scrum Team insegnando Scrum e le pratiche Agili**
- 5. protegge lo Scrum Team da interruzioni esterne e lo aiuta a crescere nel contesto aziendale**



## Development Team (DevTeam)

- 1. è un team auto-organizzato e “cross-funzionale” — ogni membro ha una specialità e tutti sanno come aiutare gli altri**
- 2. ogni membro del team decide quali feature vuole implementare (tra quelle concordate con il PO)**
- 3. ogni membro lavora su un solo progetto alla volta**
- 4. non esistono team manager e/o project manager interni**





# Scrum Team

- 1. composto da tre a nove persone**
- 2. tutti sono responsabili del lavoro svolto sul progetto**
- 3. composto da: PO + SM + DevTeam**



## Scrum Team — cosa significa “essere coinvolto”

Per far comprendere il concetto di coinvolgimento e di responsabilità dei ruoli, in Scrum si parla di “maiali” e di “polli”...

prendendo spunto da una barzelletta in cui un pollo propone a un maiale di aprire insieme un ristorante, da chiamare “uova e prosciutto”

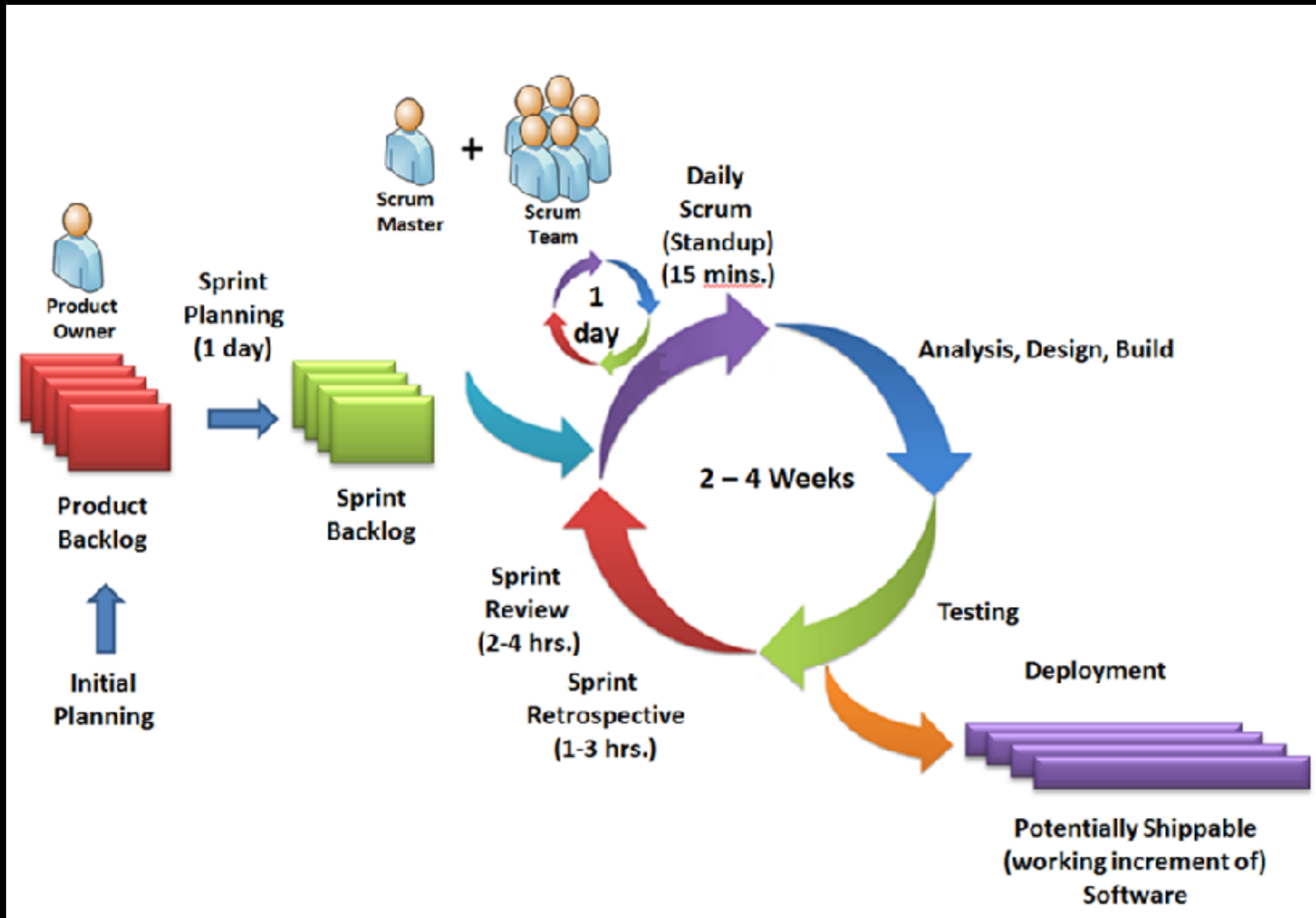
*Chiaramente, in un accordo del genere, la categoria più coinvolta è proprio quella del “maiale”...*

Fanno parte dei “pig” proprio le persone più coinvolte: Product Owner (PO), Scrum Master (SM) e il Team, cioè coloro che a vario titolo identificano e realizzano le User Story

Della seconda, categoria, quella dei chicken, fanno parte gli Stake Holder, i manager e qualsiasi altra persona che si può ritenere un “osservatore interessato” del progetto a qualsiasi titolo, ma che non si occupa in alcun modo di realizzare le storie utente

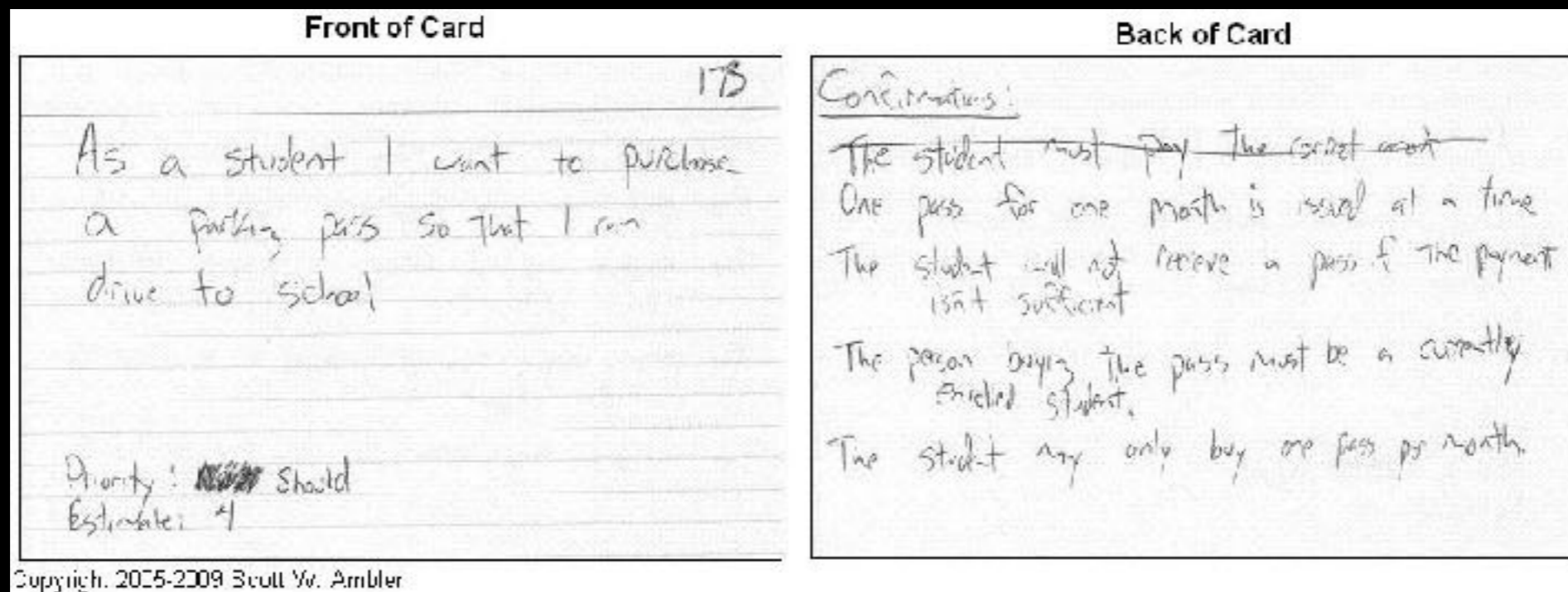


# Scrum: il processo



Possiamo definire la User Story con termini più familiari quali "requisito utente" o "funzionalità" o "specifica"

Sono tutte parole abbastanza adeguate ma non catturano pienamente lo "spirito" che Scrum assegna a una User Story



Copyright. 2015-2009 Scott W. Ambler





# Scrum in pratica — User Story

The screenshot shows a JIRA Scrum board for a project named 'Scrum: Teams in Space'. The board is divided into columns: '2 To Do', '3 In Progress Max 3', and '1 Code'. The 'To Do' column contains several user stories, including:

- TIS-8**: Requesting available flights is now taking > 5 seconds. Priority: High. Estimate: 1.5. Assignee: Emma Paris.
- TIS-68**: Homepage footer uses an inline style - should use a class. Priority: High. Estimate: None. Assignee: Jennifer Evans.
- TIS-49**: Draft network plan for Mars Office. Priority: High. Estimate: 1.5. Assignee: Harvey Jennings.
- TIS-67**: Developer Toolbox does not display by default. Priority: High. Estimate: None. Assignee: Jennifer Evans.
- TIS-20**: Engage Saturn Shuttle Lines for group tours. Priority: High. Estimate: 1.5. Assignee: Ryan Lee.
- TIS-17**: Engage Saturn's Rings Resort as a preferred provider. Priority: High. Estimate: 1.5. Assignee: Ryan Lee.

○ Story ID: \_\_\_\_\_ Story Title: \_\_\_\_\_

**User Story:**

As a: <role>  
I want: <some goal>  
So that: <some reason>

**Importance:**

**Estimate:**

**Acceptance Criteria**

And I know I am done when:

**Type:**

- Search
- Workflow
- Manage Data
- Payment
- Report/View

**Descrivere una storia utente è molto complicato e spesso la sua identificazione può essere difficile**



**Descrivere una storia utente è molto complicato e spesso la sua identificazione può essere difficile.**

**Un buon punto di partenza per capire di cosa stiamo parlando è il modello INVEST**

**Independent**

**Negotiable**

**Valuable**

**Estimable**

**Small**

**Testable**

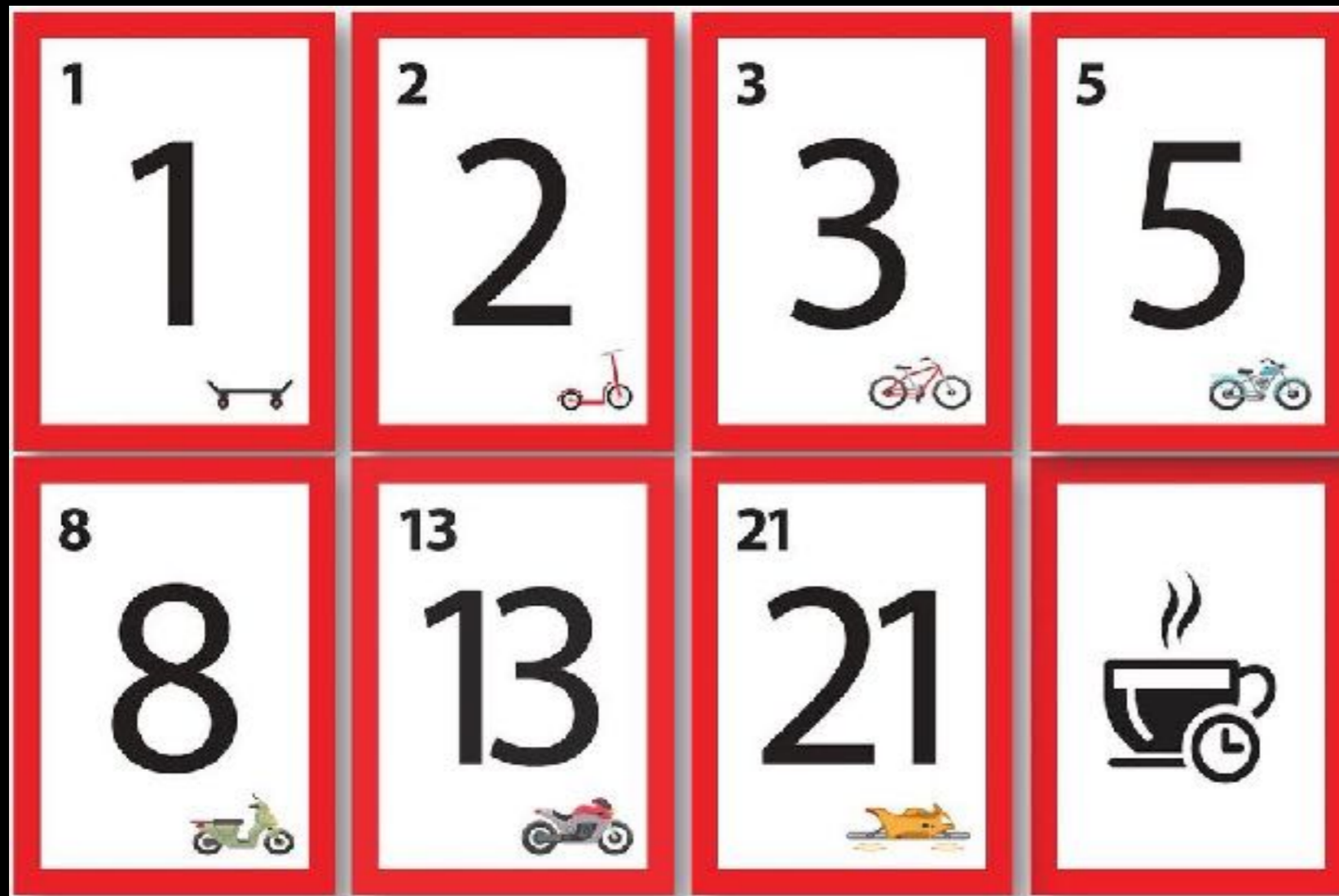


una User Story quindi dovrà soddisfare una serie di condizioni:

1. Non deve dipendere da nessuna altra storia (**Independent**)
2. Deve essere possibile chiarirla in tutti i suoi aspetti (**Negotiable**) attraverso il dibattito fino a quando tutti gli attori coinvolti non concordano sul suo contenuto
3. Esiste solo se anche l'utente finale trae vantaggio (**Valuable**) dalla sua realizzazione e
4. deve essere possibile effettuarne una stima (**Estimable**)
5. Essa deve inoltre essere piccola (**Small**)
6. e oggettivamente verificabile (**Testable**)



# Scrum in pratica: Planning Poker



metodo più  
comune di  
stima

"gioco di carte"  
per discutere,  
giustificare e  
valutare  
diverse stime  
per la  
realizzazione  
del lavoro

effettuate da  
tutti i membri  
del team per  
arrivare ad una  
stima condivisa





**Le iterazioni di un processo basato su Scrum sono chiamate Sprint che ha le seguenti caratteristiche:**

- 1. ha un obiettivo dichiarato (goal) espresso in modo chiaro e non tecnico**
- 2. ha una durata prefissata (generalmente di 2 o 4 settimane) e quindi una data di consegna (demo) ben identificata**
- 3. deve portare alla realizzazione completa di una lista di storie presenti nel Product Backlog.**



**durante la fase di "negoziazione", si devono identificare tutti gli ostacoli che possono impedire o rallentare la realizzazione di una storia**

**quando una storia viene identificata, viene registrata e catalogata in un artefatto specifico di Scrum**

### **il Product Backlog**

*ad ogni storia viene associata una "importanza" o una priorità in modo tale da poter decidere quali realizzare prima e quali dopo*



# Scrum in pratica — Product Backlog



**Stakeholders**

**Scrum Team**

**Product Backlog**



# Sprint in pratica — Backlog

**JIRA Software**

**Teams in Space**  
Scrum Teams in Space

**Backlog** Configure u

QUICK FILTERS: Product Recently updated Only my issues Server UI

**VERSIONS**

- EPIGS
- All issues
- SeeSpaceEZ Plus
- Large Team Support
- Space Travel Partners
- Summer Saturn Sale
- Afterburner Plus
- Local Mars Office
- Hyperspeed shuttles
- New launch platforms
- Delicious Space Nutrition
- Spacatainment

**Sprint 1** 14 issues 3 8 5

**Sprint 2** 6 issues Start sprint

Start: 10 Aug 2015 — Release: 9 Oct 2015

**Backlog** 49 issues Create sprint

Issue Key	Issue Title	Team	Count
TIS-25	Engage Jupiter Express for outer solar system travel	SeeSpaceEZ Plus	5
TIS-37	When requesting user details the service should return prior trip info	Large Team Support	1
TIS-9	After 100,000 requests the SeeSpaceEZ server dies	Local Mars Office	1
TIS-7	500 Error when requesting a reservation	Large Team Support	1
TIS-10	Bad JSON data coming back from hotel API	Space Travel Partners	5
TIS-18	Enable Speedy SpaceCraft as the preferred individual transit provider	Large Team Support	1





**ad ogni attività viene assegnata una durata  
certa entro limiti di tempo ristretta**

**ci si concentra solo sulle  
cose realmente necessarie**

**la consegna delle storie avviene con una  
frequenza molto alta**

**in tal modo si possono ricevere dei riscontri  
rapidi ed eventualmente correggere subito  
le cose che non vanno bene**



**riunione (in piedi) tra SM e Team**

**di solito al mattino**

**dura 15 minuti al massimo**

**ogni persona deve risponde alle domande:**

- 1. cosa ho fatto ieri?**
- 2. cosa farò oggi?**
- 3. quali sono gli ostacoli che ho rilevato?**



**Alla fine di uno sprint c'è la demo**

**dura una giornata (in alcuni casi anche qualche decina di minuti)**

**il Team presenta al PO e agli Stakeholder i risultati ottenuti**

**si parte dallo Sprint Backlog e per ogni Story si dichiara se è stata realizzata in toto o in parte (le storie non complete verranno presentate alla prossima demo)**

**in questa fase è importante raccogliere il maggior numero di riscontri per capire quali correzioni fare**



**Dopo la demo lo Scrum Master incontra il Team per ragionare insieme sui problemi che ostacolano il lavoro**

**la riunione non dovrebbe durare più di 4 ore**

**spesso viene prodotto un documento di sintesi *(sintetico appunto)***

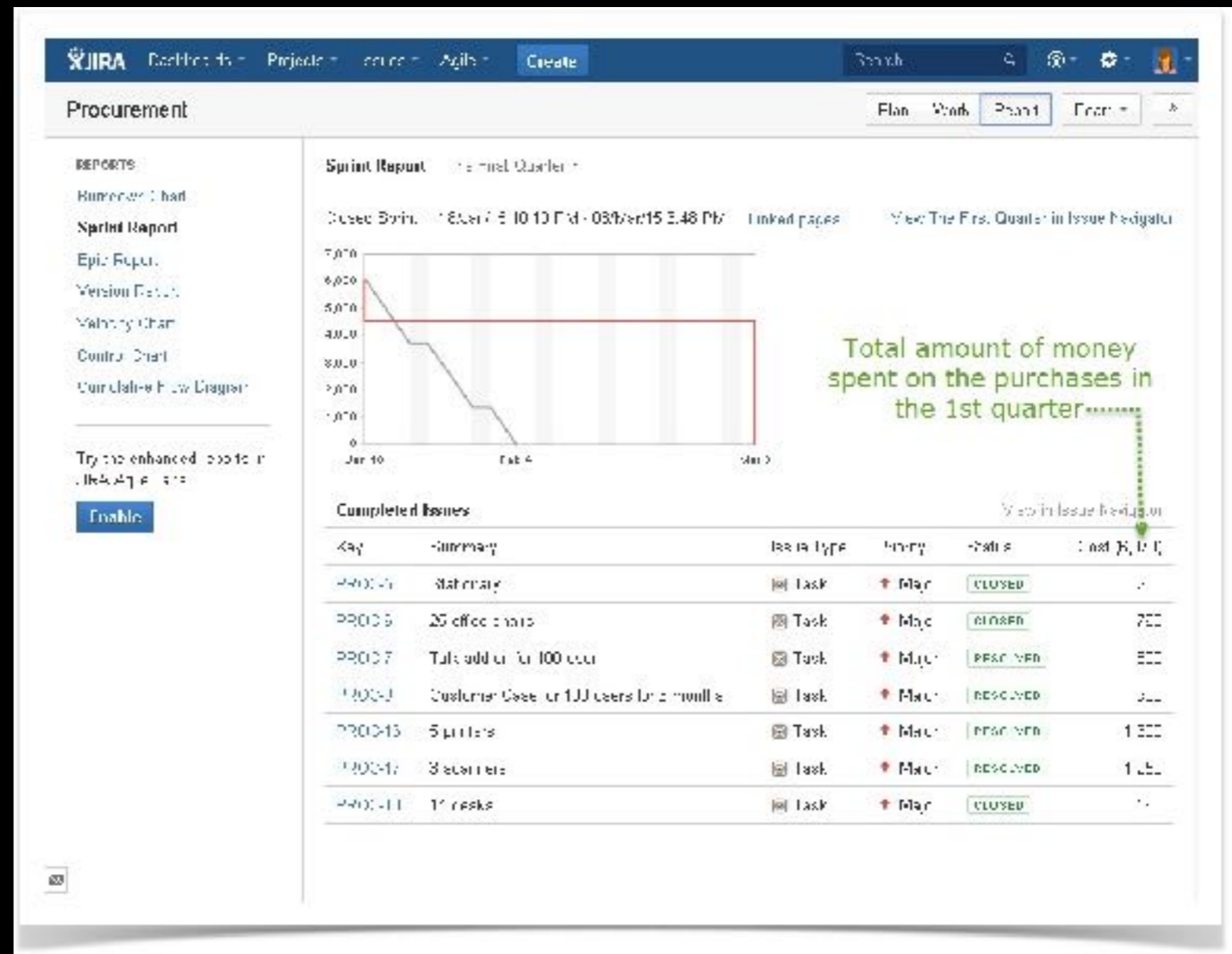
**è altrettanto spesso possibile che siano aggiornate le linee guida del progetto**





# Scrum in pratica — Sprint Monitoring

operazione  
spesso  
molto  
complicata e  
difficile da  
formalizzare  
  
si ricorre a  
strumenti  
specifici



“capire cosa sta succedendo”



la velocity di un Team è l'impegno esprimibile durante uno Sprint

ad esempio per un Team di 5 persone e uno Sprint di 3 settimane la "velocità" sarà di

**5 x 15 = 75 (giorni di lavoro per Sprint)**

velocità teorica, perché i giorni lavorativi reali saranno quasi sempre di meno: imprevisti personali, sforamenti delle stime, ...

di solito si considera la velocità media (realistica) pari a circa il 60% del valore calcolato

la velocità si può misurare anche in "story points" oppure ore, anziché giorni di lavoro, dipende dalle esigenze

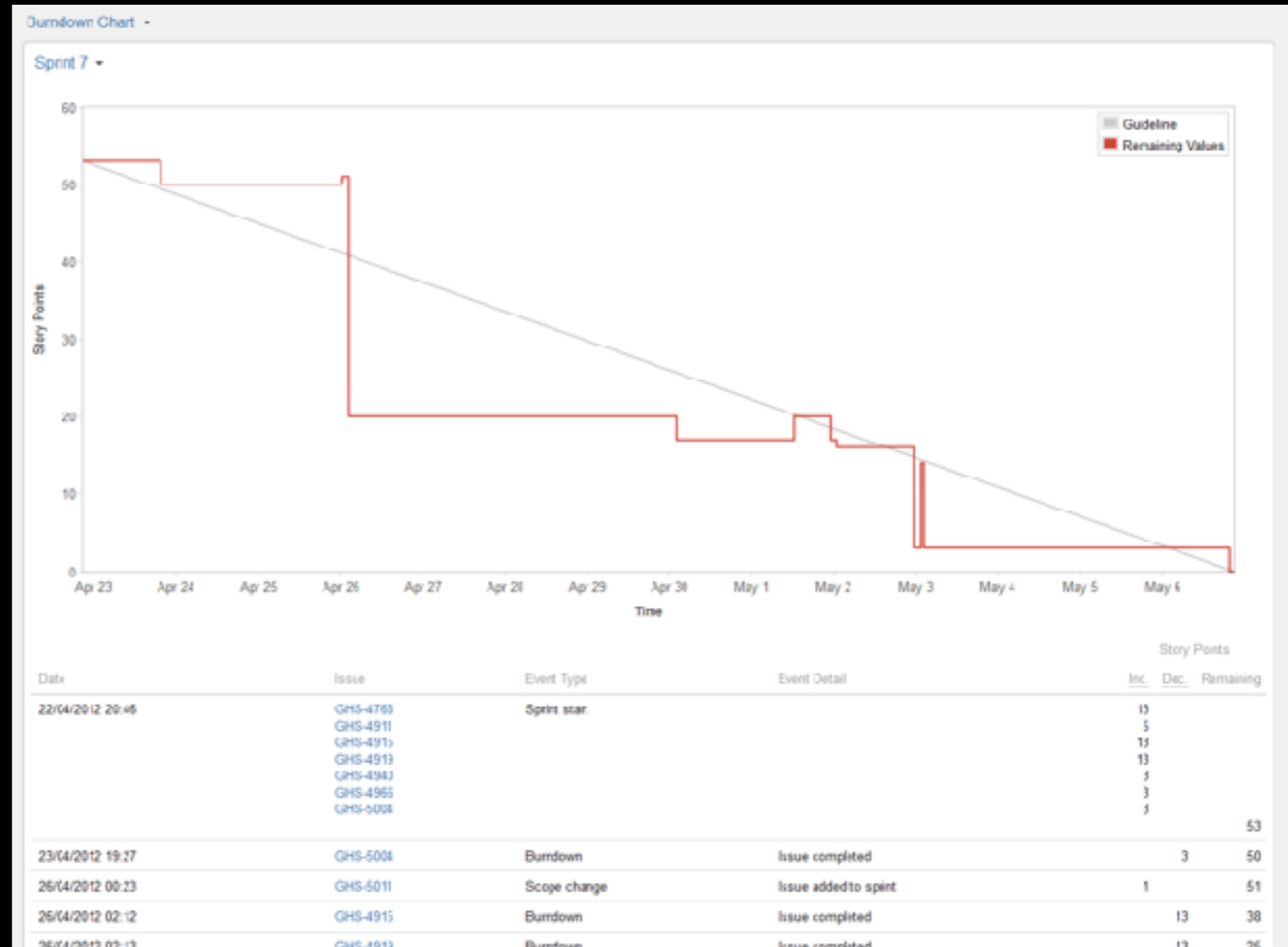


# Scrum in pratica — Burndown Chart

diagramma  
cartesiano per  
monitorare  
l'andamento di  
uno Sprint

sull'asse X sono  
indicati i giorni  
dello Sprint

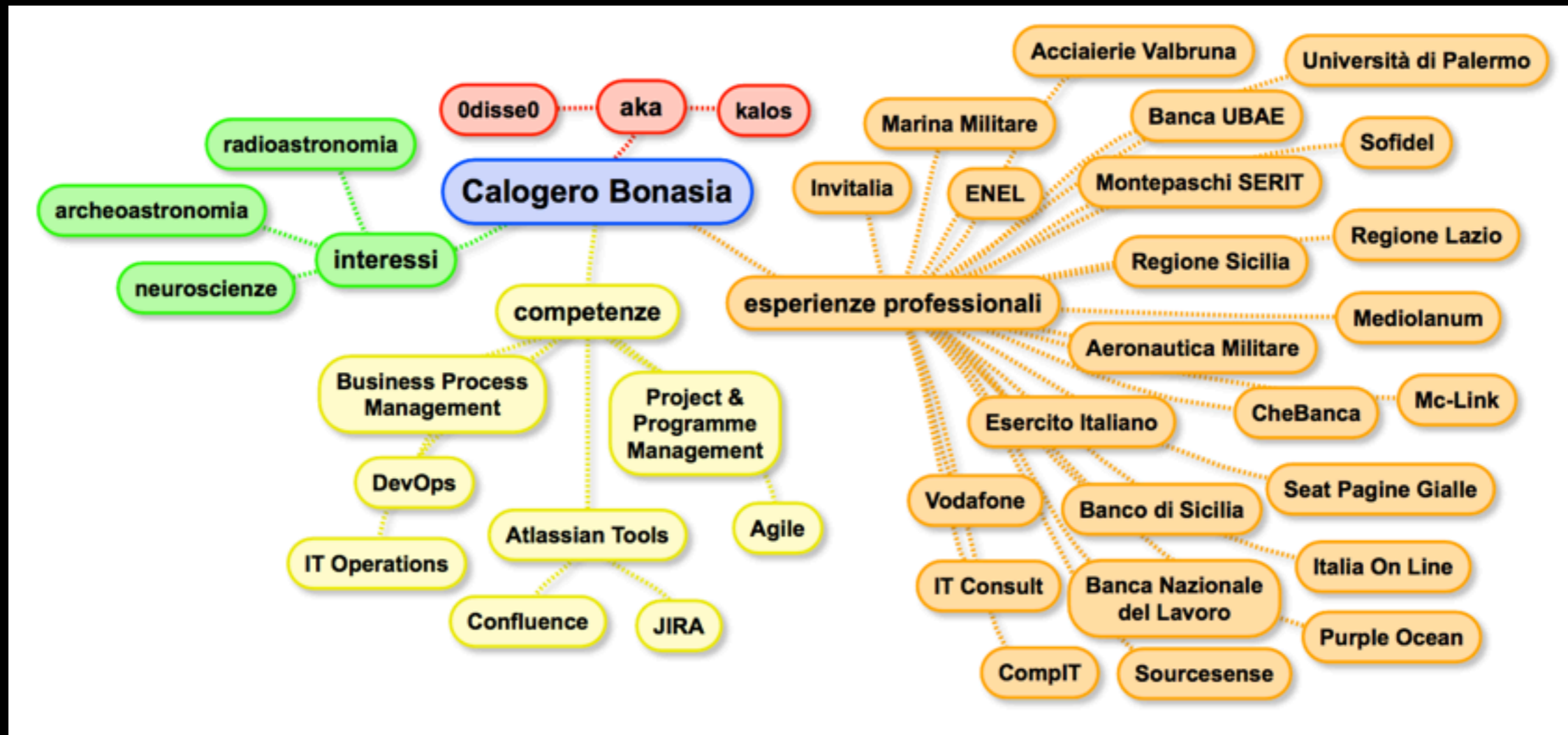
sull'asse Y sono  
indicati i giorni  
di effort



Se la linea dei giorni che mancano alla consegna è al di sopra della  
linea ideale, vuol dire che il progetto è in ritardo e quindi sarà  
necessaria una qualche analisi per adottare le adeguate correzioni.



# Calogero 'Kalos' Bonasia



# Letture Consigliate



<http://amzn.to/2itk8gV>



<http://amzn.to/2iugdQY>



<http://amzn.to/2hYuuS8>



<http://amzn.to/2hYl48Y>

