

**Modulo Master:**  
**SCRUM framework**

Docente: Francesca Petrelli

# Programma

## Modulo SCRUM framework

### SEZIONI TEORICHE

1. Recap prassi agile e connessioni tra Agile PM e SCRUM
2. SCRUM framework
  - ❑ I ruoli
  - ❑ Gli eventi
  - ❑ Gli artefatti

### SEZIONI LABORATORIALI

- Simulazione esame SCRUM Master

# SCRUM Master™ Certification

## INFORMAZIONI ESAME

1. La credenziale è rilasciata su base volontaria da International SCRUM Institute™ - ISI
2. In modalità autonoma e individuale, potete prenotare e acquistare il codice univoco per svolgere l'esame al link: [https://www.scrum-institute.org/Register\\_Scrum\\_Certification\\_Program.php](https://www.scrum-institute.org/Register_Scrum_Certification_Program.php)
3. L'esame si svolge online sulla piattaforma ISI
4. È in lingua inglese
5. Prevede un test a risposta multipla di 50 domande in 1 h



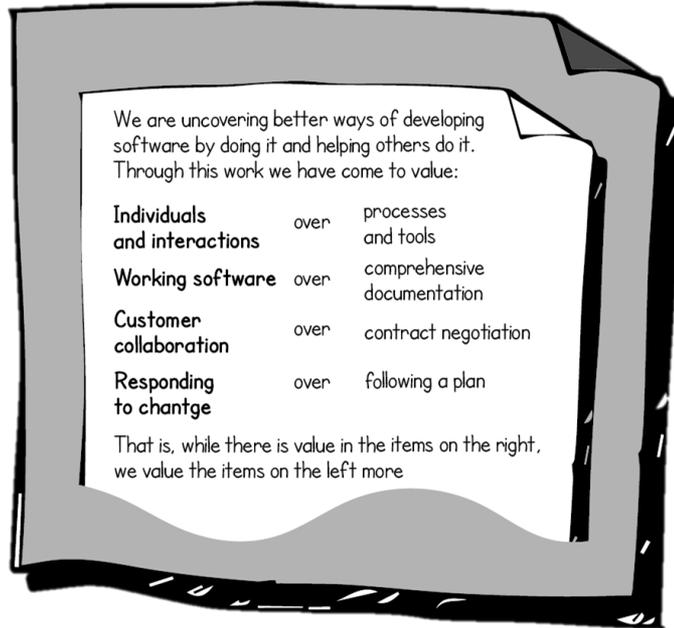
Per superare l'esame, è richiesto un punteggio minimo di almeno al 60% di risposte corrette (30)



# 1. Recap prassi principali dell'Agile way of working: principi e value driven delivery



# La base di riferimento



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

# I framework e i metodi in Agile

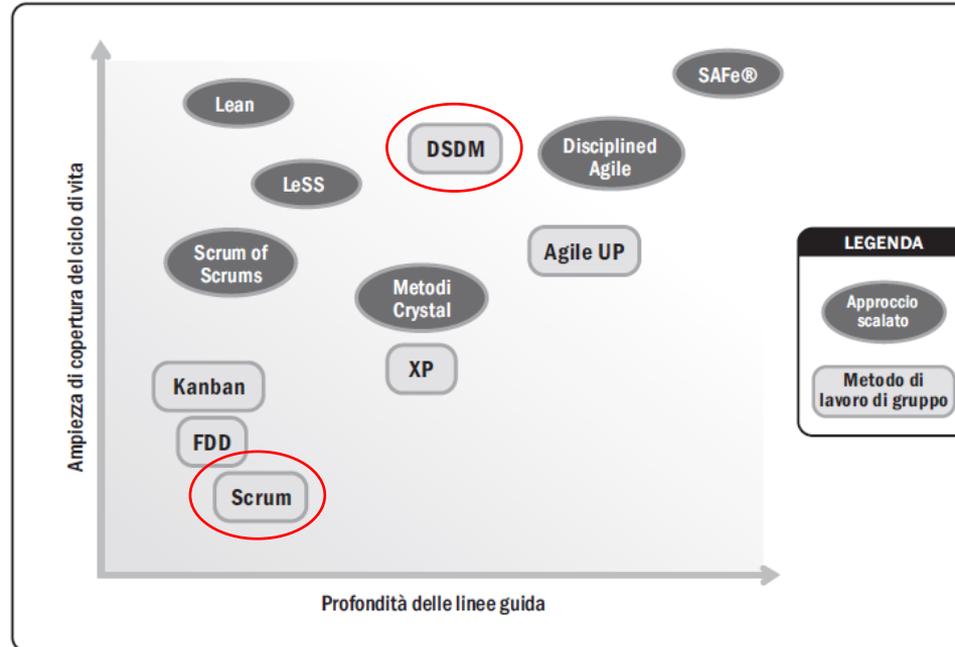
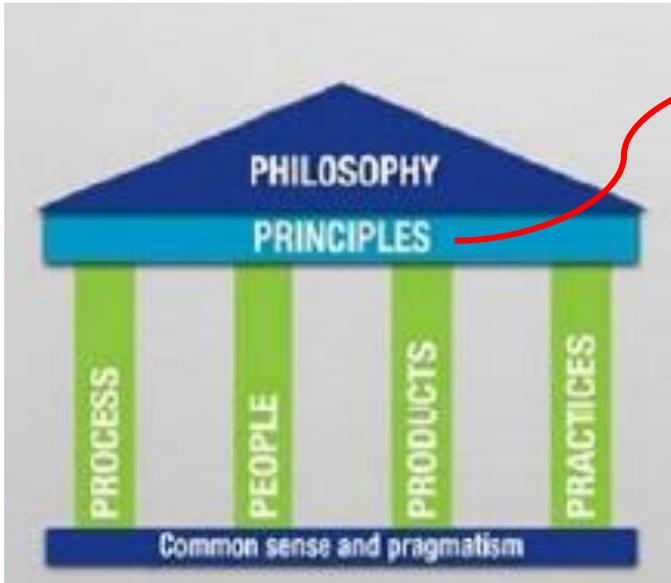


Figura A3-1. Approcci agili rappresentati per ampiezza e profondità

Agile Practice Guide, PMI, Allegato 3, p. 100.

# I principi e le tecniche DSDM sono comuni ai vari framework



FOCUS ON THE BUSINESS NEED



COLLABORATE



BUILD INCREMENTALLY FROM FIRM FOUNDATIONS



COMMUNICATE CONTINUOUSLY AND CLEARLY



DELIVER ON TIME



NEVER COMPROMISE QUALITY



DEVELOP ITERATIVELY



DEMONSTRATE CONTROL

Predittivo: Data inizio task-  
Data fine task-Durata task-  
effort RU-%avanzamento  
task

Adattivo: Amount of  
delivery wip and done;  
impediment; velocity

# 1° principio



FOCUS ON THE  
BUSINESS NEED

Il 1° principio implica:

La prioritizzazione dei requisiti (vedi prodotto: Prioritised Requirements List, elaborato ad alto livello al termine della fase Foundations) è centrale ed applicata mediante tecnica MoSCoW

Il “Minimum Usable Subset” rappresenta proprio l’output dei requisiti “Must have”.



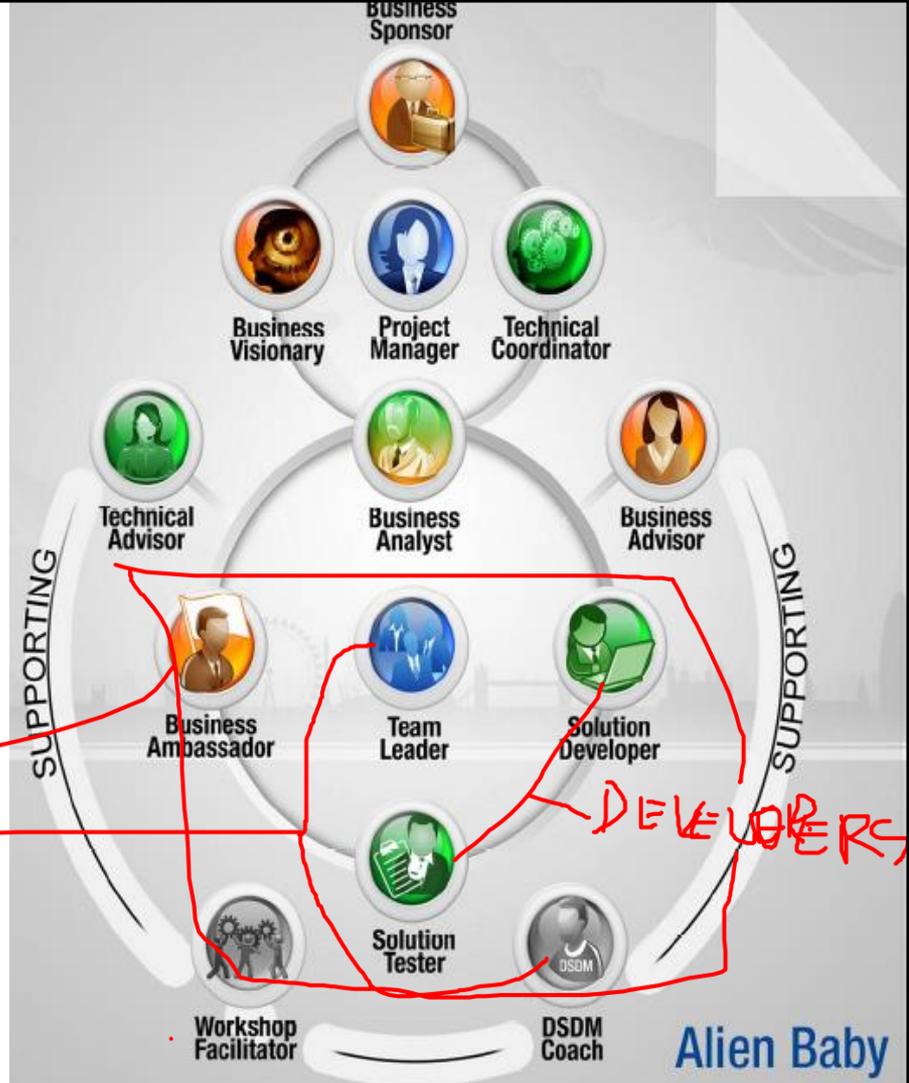
## 2° principio



COLLABORATE

implica:

- Identificare nella struttura DSDM, i tre diversi tipi di responsabilità e rispettivi ruoli: i *business roles*, quali **Business Sponsor**, **Business Visionary** e **Business Ambassador** sono i rappresentanti delle esigenze business e garanti del “focus on needs”.
- I ruoli e responsabilità in SCRUM sono di meno e meno strutturati rispetto a questi: perché secondo voi?
- **Quali ruoli potrebbero non esistere in SCRUM e perché?**
- **No PM, No TL**
- sì quelli arancio, ma PO «Product Owner»
- al posto di DSDM coach=**SCRUM Master** agisce da facilitator



## 3° principio



BUILD INCREMENTALLY  
FROM FIRM FOUNDATIONS

sottintende:

- il ciclo di vita DSDM, stabilendo una comprensione condivisa durante le fasi di Feasibility e Foundations prima di creare il prodotto incrementalmente nelle successive fasi.
- La delivery incrementale ha l'obiettivo di ottenere benefici rapidamente, ad ogni rilascio, in quanto l'"increment" è già usabile.
- La delivery incrementale crea quindi anche confidenza nel business sull'utilità nel proseguire con i successivi incrementi e sull'efficacia del metodo stesso.

Ogni iterazione SCRUM, produce un increment che viene affinato incrementalmente, SCRUM tuttavia non si occupa di gestire un progetto, dunque non esistono fasi definite del ciclo di vita, ma solo successive iterazioni incrementali

## 4° principio



COMMUNICATE  
CONTINUOUSLY AND  
CLEARLY

implica:

il ricorso in DSDM ai Facilitated Workshops che incoraggiano la collaborazione e l'efficace comprensione tra le diverse figure coinvolte

KOM → PLANNING

DAILY → // //

REVIEW → // //

RETROSPECTIVE → // //

In SCRUM, la comunicazione osmotica è altrettanto centrale: I meeting diventeranno i 4 eventi chiave di ogni timebox.

**Ricordate per DSDM il “Daily Stand up”, tenuto quotidianamente per l’avanzamento dello stato della delivery dal Solution Development Team e gestito dal Team Leader?**

In SCRUM è identico ed è gestito dallo SCRUM Master, come ogni altro evento

# 5° principio

“never compromise quality”:

- La progettazione, la realizzazione e il test della soluzione incrementale sono basati sui *needs* e le priorità concordate con il business: non ci sono sorprese allo **UAT!**
- I test stessi sono scritti durante il «refinement» degli incrementi e basati sulle DoD.
- La validazione è periodica e ad elevata frequenza, proprio per ri-calibrare la delivery sul *customer feedback*



NEVER COMPROMISE  
QUALITY

In SCRUM:

- DoD (Definition of Done)
  - ~~MoSCoW sulle DoD~~
  - review meeting
- diventano cardinali per gestire e garantire la qualità!

## 6° principio – 1/2

In DSDM, c'è “**Structured Timebox**” che contiene tre principali fasi

- **Investigation (tasks plan)**
- **Refinement (build and test)**
- **Consolidation (UAT – review and retrospective).**



DEVELOP  
ITERATIVELY

Precedute da un breve KoM per la Pianificazione del timebox e concluse da un Close-Out, per l'accettazione formale dei deliberables appannaggio del Business Visionary e del Technical Coordinator (formal timebox review).

MAX 6 WEEKS



## 6° principio – 2/2



- DSDM, descrive però anche un tipo di timebox chiamato “**free format Timebox**”, che può essere utilizzato quando gli approcci allo sviluppo sono proprio **SCRUM** o quando si necessita di una minore strutturazione del timebox.
- Il concetto di iterazione “timebox” è ciò su cui basa il ciclo di vita di progetto!
- Tale concetto diventa in SCRUM il concetto cardine o l’evento “**SPRINT**”.
- **Il timebox in SCRUM è dunque più vicino a un “free format”, ma è cadenzato da cerimonie (meeting) e artefatti (input-output) ben precisi e codificati!**



## 2. SCRUM Framework: ruoli – eventi – artefatti

# Scrum, chi è costui?

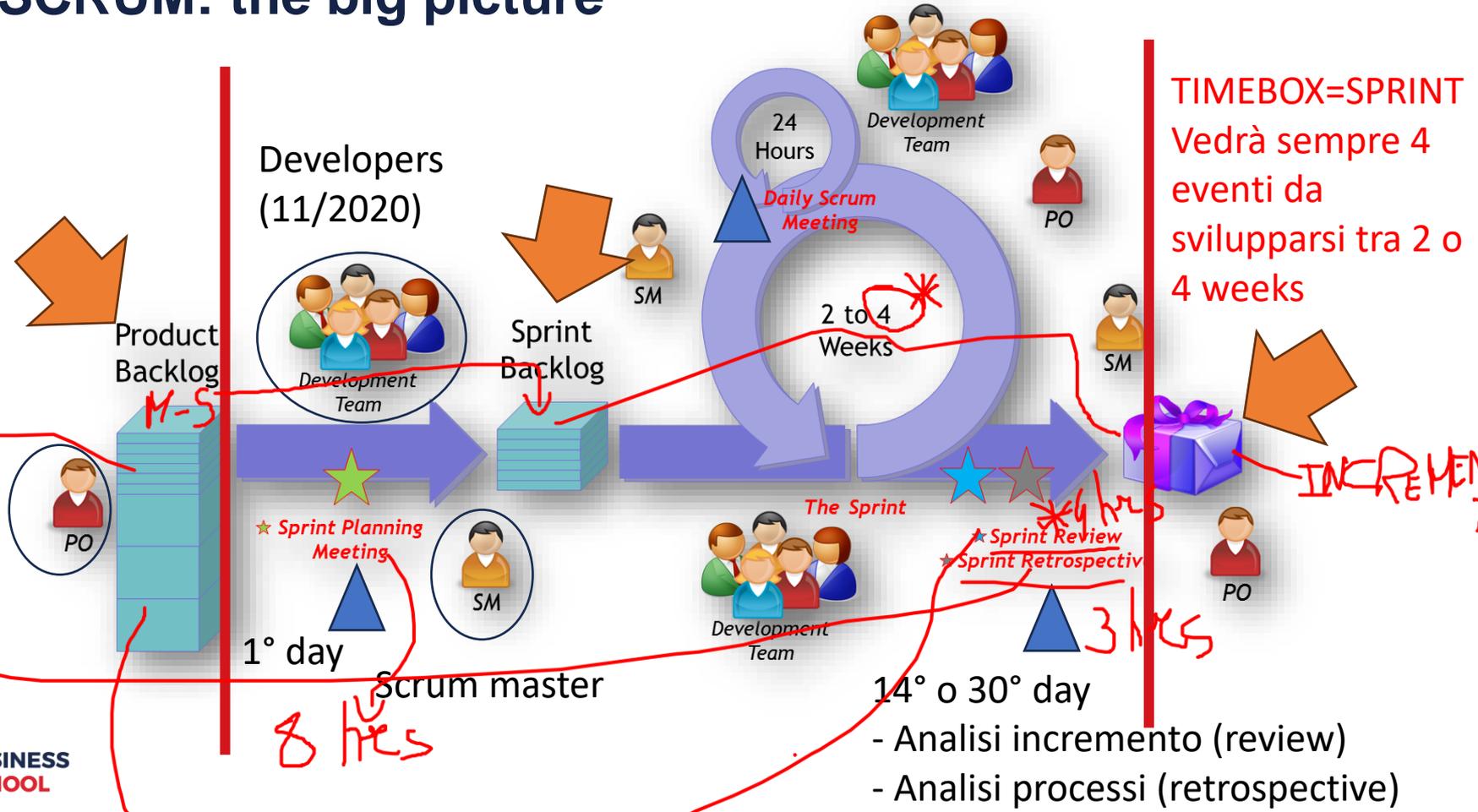
- Framework sviluppato e mantenuto da Schwaber and Sutherland (1995)
- Nasce in ambito IT per sviluppare e sostenere la realizzazione di prodotti complessi
- È caratterizzato da un insieme di **RUOLI – EVENTI – ARTEFATTI e GROUND RULES**
- Si fonda tre **core values**:
  - **TRANSPARENCY**
  - **INSPECTION**
  - **ADAPTATION**

**TRANSPARENCY** = L'obiettivo di quest'area è quello di dotarsi di un linguaggio comune e di basare la comunicazione su termini e concetti condivisi, come «value», «done» ecc.

**INSPECTION** = verificare la capacità del processo di sviluppo di massimizzarne il valore, iterando frequentemente l'analisi di eventuali scostamenti e prestazioni inadeguate

**ADAPTATION** = individuare azioni correttive rispetto al processo al fine di rendere accettabile il prodotto, cercando di rendere rapidamente implementabili le azioni correttive

# SCRUM: the big picture

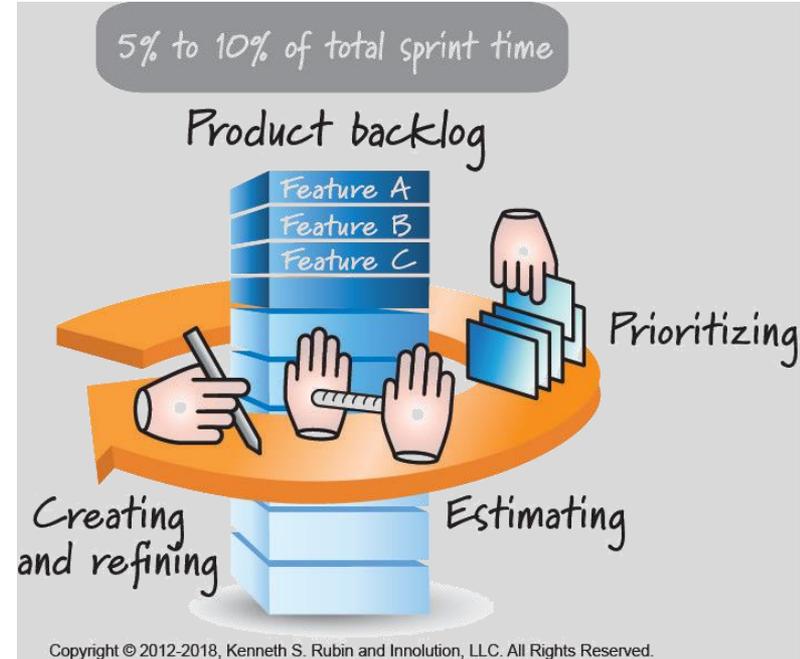


# L'artefatto «vivente»: Product Backlog

## Grooming Backlog

Le attività di affinamento del backlog includono:

- ✓ Scomposizione progressiva delle epics (business requirements di alto livello) in user stories (stakeholder requirements)
- ✓ Creazione di nuove user stories in risposta a nuove esigenze utente che man mano emergono
- ✓ Prioritizzazione delle user stories che progressivamente entrano in coda di backlog
- ✓ Ri-prioritizzazione degli item del backlog quando si popola di nuove user stories
- ✓ Individuazione delle user stories **Must Have** (prioritarie) che dovrebbero entrare nella successiva iterazione



# Lo Sprint Goal – 1/2

- Creato durante lo Sprint Planning
- È una guida sul «perché» si sta realizzando l'incremento
- I PBI selezionati per lo sprint sono implementati coerentemente con lo Sprint Goal.
- I Developers hanno sempre in mente lo Sprint Goal durante lo sviluppo

- **Cosa vogliamo creare?**
- **Qual è il nostro obiettivo questa iterazione?**
- **Possiamo dividerlo?**

# Lo Sprint Goal – 2/2

Alcuni semplici esempi:

Sprint Goal non chiaro	Sprint Goal chiaro
Migliora la funzionalità del carrello della spesa.	Semplificare il processo di acquisto per consentire un aumento dei tassi di conversione.
Migliorare le prestazioni.	Ridurre il tempo di caricamento delle pagine del X%.
Nuovo segmento di mercato a bordo.	Consentire a un nuovo segmento di mercato di acquistare il servizio Y.

# Gestione dello Sprint

Durante lo Sprint:

- Non può essere messo a rischio lo **Sprint Goal**
- Deve essere rispettata la conformità alla Qualità
- PO e Developers possono **rinegoziare e chiarire il lavoro** in base a quanto si sta apprendendo

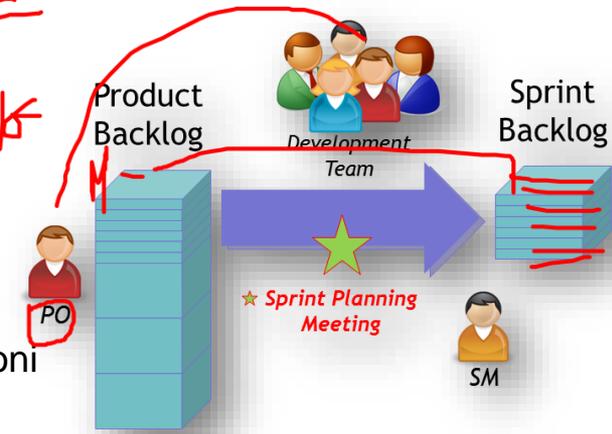
Cancellare lo Sprint:

- Solo il PO ha l'autorità per cancellare uno Sprint
- Si cancella se lo Sprint Goal diventa obsoleto
- I PBI DONE sono esaminati
- I PBI incompleti sono ristimati e reinseriti nel PB

# Sprint planning meeting – 1/2

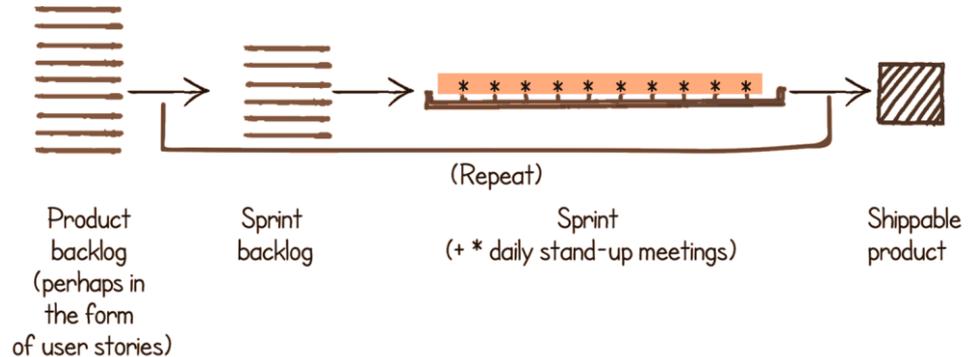
- ❑ Costituisce il momento temporale della pianificazione dello sprint stesso, in termini di:
  - ✓ Cosa va realizzato
  - ✓ Quale è l'effort necessario a raggiungere l'obiettivo
- ❑ Il planning avviene in un meeting della durata di 8 h su uno sprint di 1 mese, alla presenza di tutto lo Scrum Team.
- ❑ Prevede di:
  - ✓ Analizzare congiuntamente il Product Backlog, valutare le prestazioni precedenti, fare delle previsioni circa l'effort dello sprint
  - ✓ I Developers sono responsabili delle forecast e lo Scrum Master le valida definendo lo sprint goal, inteso come implementazione del Product Backlog.
- ❑ Product Backlog per quel timebox + il piano con l'effort e la scomposizione del lavoro, costituiscono l'artefatto denominato: Sprint Backlog

WHAT — GROOMING  
HOW — SIZING



# Area «Adaptive planning» - 1/2

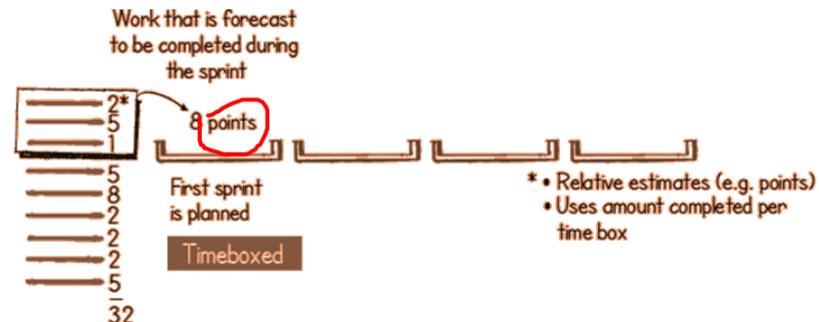
- Vincolare le deadline di una release e dei suoi singoli sotto-timebox definendone la finestra temporale in base alle capacity del team
- I timebox nello specifico framework SCRUM sono definiti come «Sprint»
- Negoziare e discutere tra «customer» e team specialistico **quali item del PB censire** nello specifico **timebox**, in base a un doppio criterio crociato di prioritizzazione: business value e technical feasibility



# Area «Adaptive planning» - 2/2

- Assegnare l'effort delle US non più in base al criterio standard di «durata», ma al concetto di **«size» = complessità tecnica relativa**
- Definire e condividere intra-team il criterio standard convenzionale per l'attribuzione delle «size» e delle relative scale, quali:
  - **«user/story points»**
  - **«T-shirt»**
- Gli story points: iniziali di una iterazione vengono progressivamente affinati in base alla velocity (capacità di performance) espressa nei primi sprint

- Ogni point è assegnato in rapporto non lineare a un point base, considerato elementare e semplice da realizzare (1 o 2 sulla sequenza Fibonacci)
- Individuare un requisito quale punto di partenza elementare e noto per l'assegnazione delle «size» che possa essere utilizzato come sistema di riferimento baseline



# Le stime agili: criteri

Per stimare le US, prendere in esame i seguenti indicatori:

- **Complessità**: difficoltà tecnica e il grado di innovatività
- **Rischiosità**: livello di esperienza del team nello sviluppo di features analoghe
- **Interdipendenze**: criticità derivanti da fattori esterni (es. Interfaccia con IT centrale o di altri Paesi, disponibilità ambienti di sviluppo e test)

**Le stime del team devono convergere in base alla reciproca persuasione!**

# Planning Poker: step-by-step

<https://planningpokeronline.com>

Cards

- Ogni membro del team ha a disposizione le carte del Planning poker secondo la serie di Fibonacci: 1, 2, 3, 5, 8, 13

User Stories

- Le user stories sono consultabili

US Start point

- Lo Scrum Master sceglie una US ben conosciuta cui si assegna valore 1 (baseline)

Size

- Si sceglie la 1° US da stimare: ogni membro sceglie la carta relativa alla propria stima senza mostrarla agli altri

Discussion

- Le carte vengono girate e discusse le motivazioni del valore più alto e più basso assegnato. Quindi si rivota

Decision

- Il valore espresso dalla maggioranza dei membri sarà preso quale stima migliore

10 US  
100 Story points

300 POINTS  
200 POINTS

# Il concetto di «Velocity»

- La velocity è il numero di story point che un team può completare in una iterazione
- Le modalità di stima della velocity sono:
  - **ricorso a dati storici (solo team stabili con storia pregressa)**
  - **iterazione di prova**
  - **Previsioni (al tempo zero)**
- dopo le prime due o tre iterazioni, il dato di velocity (a meno di impediments) resta costante e si parla di velocity driven planning

**Velocity:** rate of progress a team is making. The velocity can be used to empirically forecast their future rate of progress (assuming that the conditions remain the same).

You can know **how much work is going to be delivered by tracking overall velocity**

# Sprint planning meeting – 2/2



# Daily meeting

- ❑ Meeting di 15' interno ai Developers serve a:
  - ✓ Fare «inspection» di quanto prodotto fino al precedente D.S.
  - ✓ Definire il piano delle successive 24h
- ❑ Le domande abilitanti il D.S. sono:
  - ✓ Cosa abbiamo fatto ieri per raggiungere lo sprint goal (controllo dell'avanzamento)
  - ✓ Cosa facciamo oggi per raggiungerlo (pianificazione di dettaglio)
  - ✓ Quali eventuali impedimenti, ostacoli esistono per raggiungerlo
- ❑ Lo Scrum Master assicura che il D.S. si concluda in 15'
- ❑ L'obiettivo di questo meeting è assicurare partecipazione e commitment del team, registrando gli «impediment» rispetto al pianificato del D.S. Precedente
- ❑ Gli «impediment» saranno presi in carico dallo Scrum Master (Servant Leadership) che individuerà i correttivi o indirizzerà verso altri stakeholder le possibili soluzioni.

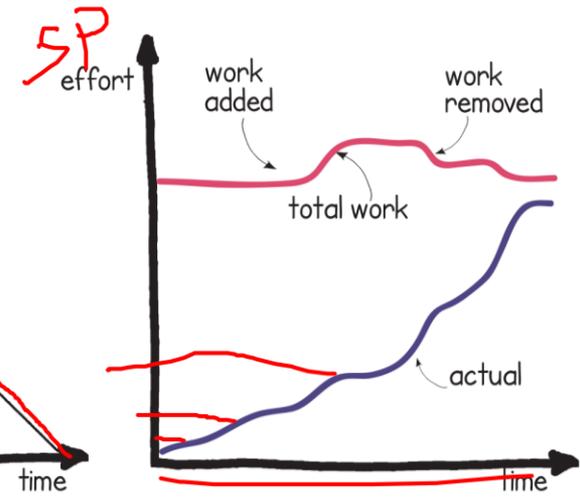
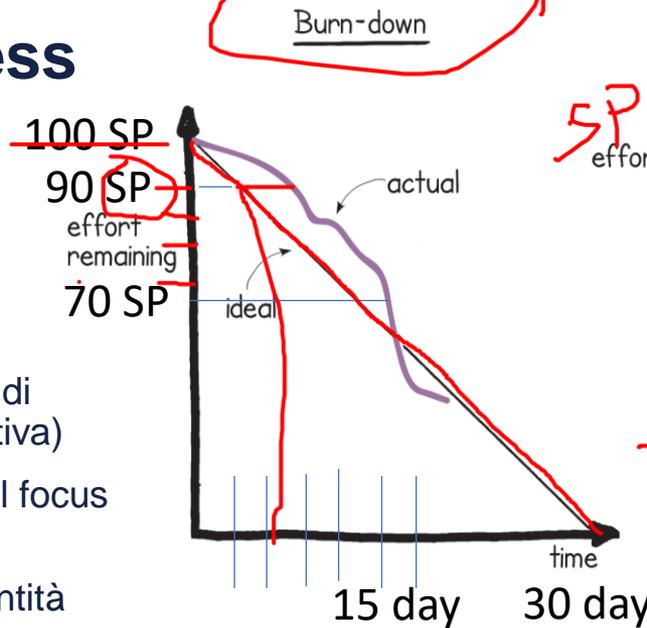


# Burn charts sul progress

Esistono due grafici di consuntivazione del delivery:

- burn-down chart
- burn-up chart

- Mostrano la situazione corrente in termini di valore di avanzamento (velocity performativa)
- In entrambe le visualizzazioni comunque il focus è su ciò che è stato completato
- **Burn-down charts** assumono che la quantità del lavoro non cambi nel tempo
- **Burn-up charts** dovrebbe essere invece utilizzato quando la quantità del lavoro è probabile che cambi



- Possono essere consegnate al termine di uno sprint solo le US done.
- Pertanto, è vietato tenere conto anche dei task incompleti (test parziali – solo scrittura codice – allestimento parte di manualistica ecc.

# Sprint review meeting

- ❑ Meeting di 4h per uno sprint di un mese, organizzato al termine dello sprint stesso, volto a:
  - ✓ Fare «inspection» di quanto effettivamente realizzato («done») rispetto agli item del Backlog previsti per lo sprint
  - ✓ Coinvolgere tutte le parti interessate per una valutazione del deliverato e per definire i probabili items del Backlog da collocare nello sprint successivo.
- ❑ Lo Scrum Master garantisce di:
  - ✓ Il Product Owner illustri il «done» e il «not done»
  - ✓ Il Product Owner spieghi l'avanzamento complessivo del P.B.
  - ✓ Il Dev. Team discuta dei successi e insuccessi occorsi durante lo sprint
  - ✓ L'intero Scrum Team valuti in che modo nel successivo sprint massimizzare il valore del prodotto realizzato
  - ✓ L'intero Scrum Team collabori per definire i successivi item da realizzare



# Product demonstration: obiettivi

DONE  
e' quando  
abbiamo un  
FEEDBACK

DONE  
DONE  
e' quando e'  
DELIVERABILE  
(TESTATO & DOCUMENTATO)

- Rappresenta il miglior status report possibile in termini di efficacia comunicativa
- Valuta «how much is being delivered», cioè «amount of delivery»
- Esamina e controlla la qualità del delivery
- Individua i benefici realizzati
- Favorisce le change requests «a valore aggiunto» e motivate dall'uso dello prodotto
- Produce il customer feedback che alimenta il value driven delivery

# Sprint retrospective meeting

- ❑ Successivamente alla Review, lo Scrum Team si riunisce – in un incontro di 3h – per riflettere insieme:
  - ✓ rispetto al processo di lavoro condotto nello Sprint
  - ✓ individuare gli interventi migliorativi da attuare nello Sprint successivo
- ❑ Lo Scrum Master garantisce di:
  - ✓ Analizzare l'andamento dello Sprint rispetto alle persone, relazioni, processi produttivi e strumenti
  - ✓ Individuare possibili azioni migliorative, capitalizzando i comportamenti virtuosi da applicare anche nel successivo Sprint
  - ✓ Formalizzare un piano di attuazione delle azioni migliorative
- ❑ L'implementazione delle azioni nel successivo Sprint costituirà «Adaptation» sul processo e sul Team condotta durante la Retrospective, al fine di migliorare sempre più la qualità del lavoro e del relativo prodotto.



# Il flusso della retrospective

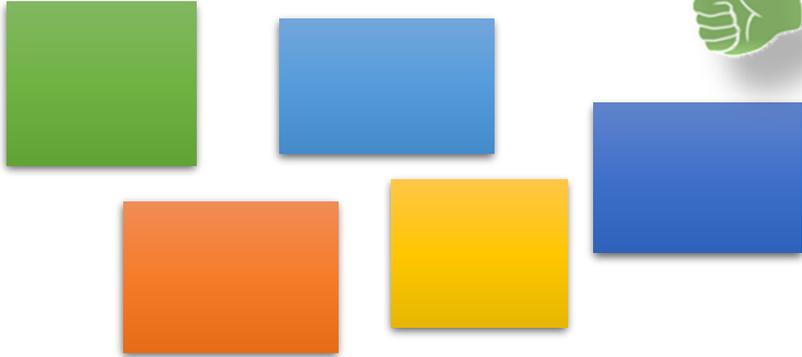
Flusso	Elementi di riferimento
Definizione dei metodi e delle regole della retrospective - <b>Set the goal</b>	<ul style="list-style-type: none"> <li>Stabilire obiettivi</li> <li>Stabilire le regole di base e la tempistica della retrospective</li> <li>Ricordare il perimetro del progetto (le baseline) e i relativi criteri prestazionali</li> </ul>
Raccogliere i dati e analizzare le fonti - <b>Gather data</b>	<ul style="list-style-type: none"> <li>Esaminare tutti i report di avanzamento e registri di progetto (<u>hard data</u>)</li> <li>Raccogliere le sensazioni e le opinioni del team (<u>soft data</u>)</li> </ul>
Discussione e individuazione aree di focus (sia critiche sia positive) - <b>Generate insight</b>	<ul style="list-style-type: none"> <li>Individuare punti di forza, di debolezza e eventi straordinari</li> <li>Analizzarne le relative cause di insorgenza</li> </ul>
Assumere decisioni - <b>Decide what to do</b>	<ul style="list-style-type: none"> <li>Individuare raccomandazioni correttive e/o migliorative</li> <li><u>Prioritizzare</u> le raccomandazioni</li> <li>Registrare i responsabili delle raccomandazioni e le tempistiche di eventuale attuazione</li> </ul>
Chiusura retrospective - <b>Close the retrospective</b>	<ul style="list-style-type: none"> <li>Revisione dell'efficacia della retrospective</li> <li>Archiviazione della documentazione di retrospective</li> </ul>

# Modello per Minute of Retrospective meeting

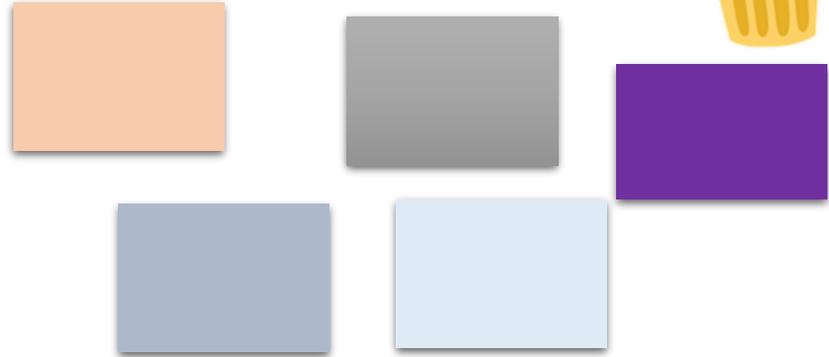
Sprint retrospective Date: *gg/mm/aa*

Team: *name of all team members*

## What went well



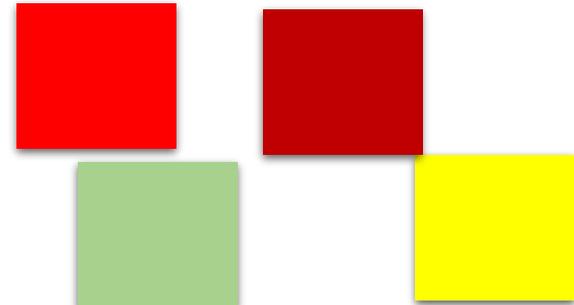
## What went wrong/poorly



## New ideas



## Action to take



### 3. Simulazione esame SCRUM Master





## **What kind of software development projects can be executed by Scrum Project Management Framework?**

- A. Complete software packages
- B. Customer projects
- C. Sub-systems, components or parts of bigger systems
- D. All kinds of software development projects
- E. None of the given answers





## What does **NOT** belong to cornerstones of the agile manifesto?

- A. Individuals and interactions over processes and tools
- B. Working software over comprehensive documentation
- C. Processes over people
- D. Customer collaboration over contract negotiation
- E. Responding to change over following a plan





---

## What is defined by the Scrum Framework?

- A. Rules & Roles
- B. Document guidelines
- C. Artifacts and events

- Choice-1: A
- Choice-2: B
- Choice-3: C
- Choice-4: A, B ,C
- Choice-5: A, C





---

## Where are the customer requirements stored?

- A. In the Product Backlog
  - B. In the Sprint Backlog
  - C. In a database
  - D. In a Scrum Product Requirement Specification
  - E. Nowhere. The Scrum Product Owner knows them
-



---

## Which ones of the following main roles are defined by Scrum Framework?

- A) Scrum Tester
- B) The Scrum Team
- C) Scrum Manager
- D) Scrum Master
- E) Scrum Product Owner

- **Choice-1:** A, B, C, D, E
  - **Choice-2:** B, C, D, E
  - **Choice-3:** B, D, E
  - **Choice-4:** A, B, D, E
  - **Choice-5:** A, B, C, D
-



---

## In software engineering what are the disadvantages of the classical waterfall model?

- A) End-Product has to be fully anticipated beforehand
- B) Some requirements are implemented as defined in the beginning of the project, and yet they are not really needed by the customer
- C) Each phase is strictly separated
- D) The scope could change

- **Choice-1:** A, B
  - **Choice-2:** B, C
  - **Choice-3:** B, C, D
  - **Choice-4:** A, B, C
-



**Q:1-Tracking project impediments in a Scrum project is whose primary responsibility?**

**Mark one answer:**

- Tester
- ScrumMaster
- Functional Manager
- Developer





**Q:4-Who is responsible for prioritizing the product backlog?**

**Mark one answer:**

- Product Owner
- Project Manager
- Lead Developer
- Business Analyst





**Q:6-What is the effect of having a large visible Story board on a wall?**

**Mark one answer:**

- It removes the need to create any other reports for management
- It continuously communicates progress within the team and to other stakeholders
- It allows the Project Manager to allocate tasks to specific team members
- It is restrictive, as it does not allow the team to innovate and change





**Q:8-Which of the following is NOT a typical artifact of the Scrum framework?**

**Mark one answer:**

- Product Backlog
- Sprint Backlog
- Burn down chart
- Gantt chart





**Q:9-How should work be allocated to the team in an Agile project?**

**Mark one answer:**

- The Team Leader (ScrumMaster) should allocate specific tasks to individuals
- Tasks should be randomly allocated to team members, using Planning Poker
- Team members should self-select tasks appropriate to their skills
- The most complex tasks should be allocated by the Team Leader (ScrumMaster)





**Q:1-Which of these may a Development Team deliver at the end of a Sprint (choose 2)?**

**Mark one answer:**

- A single document, if that is what the Product Owner asked for
- An increment of software with minor known bugs in it
- Failing unit tests, to identify acceptance tests for the next Sprint
- An increment of working software that is "done."





**Q:2-Which of the following is not a Product Owner responsibility?**

**Mark one answer:**

- Running the daily scrum meeting
- Inspecting work at Sprint Review
- Gathering requirements for Product Backlog items





**Q:3-The Sprint Goal is selected before the Sprint Backlog is created.**

Mark one answer:

True

False





**Q:4-Under what circumstances should the Product Backlog be reprioritized?**

**Mark one answer:**

- The Scrum Master should reprioritize the Product Backlog only at the end of a new Sprint.
- The Scrum Master should reprioritize the Product Backlog only at the beginning of a new Sprint.
- The Team should reprioritize the Product Backlog only at the end of a new Sprint.
- The Product Owner should reprioritize the Product Backlog whenever new information is learned.





**Q:5-What is the maximum amount of time a sprint retrospective should take?**

**Mark one answer:**

- 1 hour
- 1 and half hour
- 3 hours, for an 30 days Sprint





**Q:8-As the Sprint planning progresses, the workload has grown beyond the development team's capacity. Which action makes most sense for the Team?**

**Mark one answer:**

- Work overtime for the Sprint
- Collaborate with the Product Owner and potentially remove or change items
- Cancel the Sprint
- Star the Sprint and recruit additional team members





**Q:9-Which of the following is reflected in a Sprint Burndown Chart?**

**Mark one answer:**

- Team Members Name
- Number of Product Backlog Items Completed
- Number of Tasks Remaining
- Work Hours Remaining



**Q:10-Development Team members volunteer to own a Sprint Backlog item:**

**Mark one answer:**

- During the Daily Scrum
- Whenever a team member can accommodate more work
- Never. All Sprint Backlog Items are "owned" by the entire Development Team, even though each one maybe done by an individual team member
- At the Sprint planning meeting





**At the close of an iteration, some features may not be accepted by the customer. Features that are not accepted by the customer are:**

- A) Returned to the product backlog for re-evaluation during the next round of planning
- B) Deleted from the project
- C) Delivered to the customer but not used
- D) Free to be used on other products

