



Agile Development e Scrum

Laura Toschi
Università di Bologna

Corso Gestione dell'Innovazione e dei Progetti

CIEG – Università di Bologna



Contenuti della lezione

- ✓ Agile Development: origine e caratteristiche principali
- ✓ Varianti di Agile Development
- ✓ Approccio Scrum allo sviluppo prodotti/sw
 - Principi base
 - Ruoli
 - Eventi
 - Tecniche (artefatti)

CIEG – Università di Bologna



Agile Development: origini e caratteristiche principali

CIEG – Università di Bologna



Agile Project Management

- Il **rischio** dei modelli tradizionali di gestione dei progetti è quello di seguire un **approccio rigidamente sequenziale**. Nei casi di forte dinamismo e elevata incertezza di tecnologia/mercato può essere controproducente, perchè i requisiti sono poco chiari o cambiano continuamente.
- Dalla fine degli anni '90, per affrontare queste sfide, emergono modelli di gestione dei progetti di sviluppo prodotti (soprattutto software) di tipo **Agile (Agile Development)**
- Gli approcci Agili si fondano su un **modello di sviluppo flessibile, iterativo e incrementale**, nel quale i requisiti e le soluzioni si modificano e si adattano ai cambiamenti esterni grazie al lavoro collaborativo del team.

4

“Manifesto for Agile Software Development”



- Febbraio 2001: viene fondata l'Agile Alliance con la pubblicazione del “Manifesto for Agile Software Development” (www.agilemanifesto.org):

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan”



Elementi principali approcci Agili



1. Ricorso a **team inter-funzionale** con ampie responsabilità.
2. Frequenti **iterazioni di progettazione e crescita incrementale** del prodotto.
3. **Sperimentazioni** estensive e anticipate.
4. **Coinvolgimento sistematico** e continuo di **clienti e utilizzatori**.
5. **Pianificazione snella** e flessibile e **controllo visuale** avanzamento.
6. **Frequenti momenti di controllo** del progetto (es. daily builds).
7. Un'architettura di prodotto **modulare e “aperta”**.

CIEG – Università di Bologna

Iterazioni e accrescimento funzionale



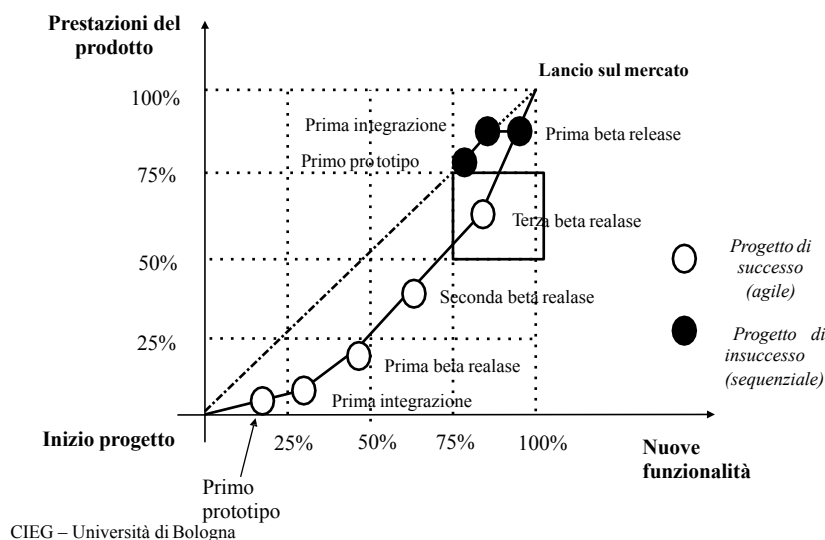
Il progetto prende forma in modo **evolutivo**, per **accrescimento funzionale e prestazionale**, a partire dal prototipo iniziale fino alla configurazione finale del prodotto.

Passaggi chiave (per progetto sviluppo sw)

- ✓ Il momento in cui per la prima volta è mostrato un prototipo, anche non funzionante, ai clienti
- ✓ Il momento in cui per la prima volta i diversi moduli funzionali del sw sono integrati tra di loro (compilati)
- ✓ Il momento in cui per la prima volta i clienti di riferimento hanno potuto utilizzare un prototipo funzionante del sw (beta release)

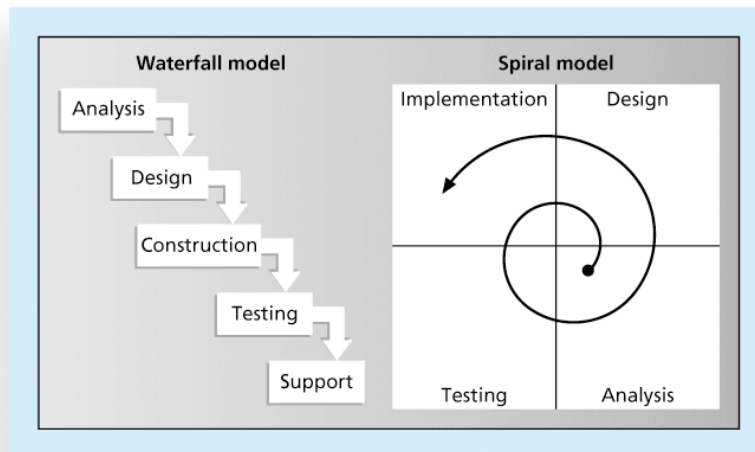
CIEG – Università di Bologna

I benefici di iterazioni e apprendimento nello sviluppo software (Iansiti et al., 2002)



CIEG – Università di Bologna

Modelli di sviluppo sequenziale (Waterfall) e iterativo (a spirale)



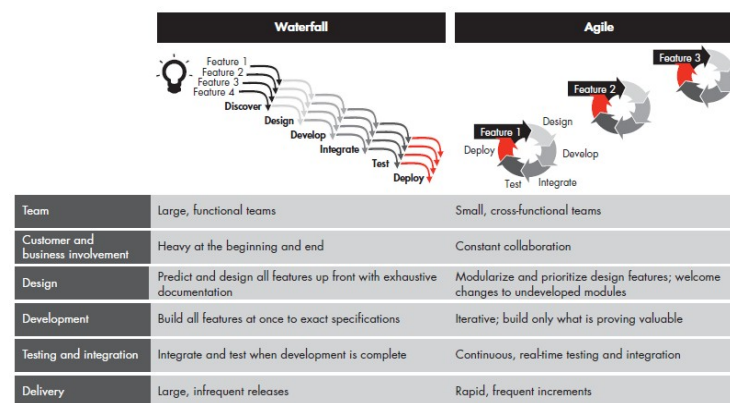
Fonte: Schwalbe (2008) Information Technology Project Management, Eighth Edition

9

Modello Sequenziale vs Modello Agile



Agile innovation integrates design with development, adapting and releasing the most valuable features far faster



Source: Bain & Company

10

Source: Bain & Co "Agile Innovation"

Quando funzionano i modelli Agili?



Conditions favorable to Agile vs. traditional methods

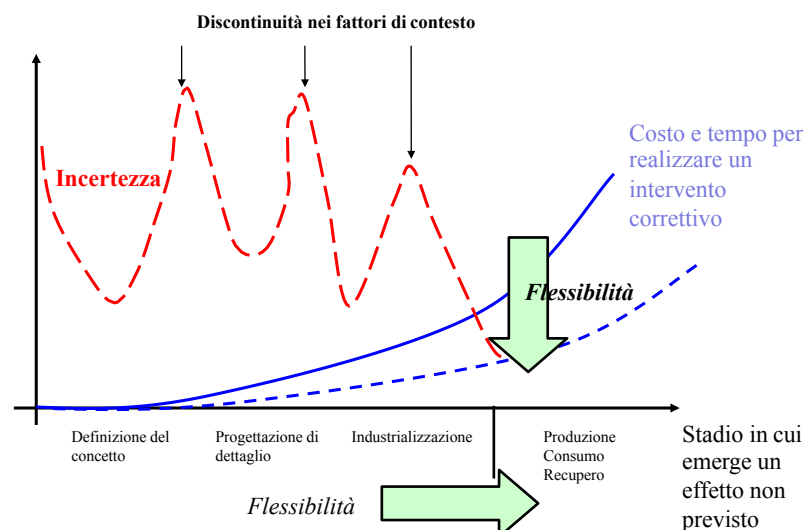
Conditions	<div style="display: flex; align-items: center;"> <div style="width: 50%; text-align: center;">← Favorable to Agile</div> <div style="width: 50%; text-align: center;">Favorable to traditional →</div> </div>	
Market environment	<ul style="list-style-type: none"> Customer preferences and solution options change frequently 	<ul style="list-style-type: none"> Market conditions are stable and predictable
Customer involvement	<ul style="list-style-type: none"> Close collaboration with customers; rapid feedback from customers is feasible; customers learn more about what they want as the process progresses 	<ul style="list-style-type: none"> Customer requirements are clear at the outset and will remain stable throughout the process; customers are not available for constant collaboration
Innovation type	<ul style="list-style-type: none"> Problems are complex, and solutions are unknown; scope is not clearly defined; product specifications may change; creative breakthroughs and time to market are important; interactive, cross-functional collaboration is vital 	<ul style="list-style-type: none"> Similar activities have been done before; innovators believe solutions are clear; detailed product specifications and work plans can be forecast with confidence; conformance to specifications is important; problems can be solved sequentially in functional silos
Modularity of work	<ul style="list-style-type: none"> Incremental developments have value and can be used by customers; work can be modularized and conducted in rapid, iterative cycles; late changes are manageable 	<ul style="list-style-type: none"> Late changes are expensive or impossible; customers cannot start testing parts of the product until everything is complete; the minimum viable product is the fully completed product
Impact of interim mistakes	<ul style="list-style-type: none"> Interim mistakes provide valuable learning opportunities 	<ul style="list-style-type: none"> Interim mistakes can be catastrophic
Corporate culture	<ul style="list-style-type: none"> The culture is team-oriented, collaborative, innovative and eager to delegate; employee turnover is relatively low 	<ul style="list-style-type: none"> The culture relies on top-down direction and functional specialization; employee turnover is high, and mutual trust is low

Source: Bain & Company

11

Source: Bain & Co "Agile Innovation"

Il contesto per l'approccio agile allo sviluppo nuovi prodotti



CIEG – Università di Bologna



Agile Development: varianti

CIEG – Università di Bologna



Le varianti dell' Agile Development

- Esistono una **molteplicità di varianti** di Agile Development, tanto che sarebbe meglio parlare di approcci Agili. Tra queste:
 - Lean Development
 - Kanban
 - Scrum



Lean Development



- Il Lean Development, in ambito sviluppo software e nuovi prodotti, riprende e adatta concetti chiave dell'approccio **Produzione Snella** (Lean Manufacturing), nato in Toyota e Honda in Giappone.
- Il cuore dell'approccio si concentra sulla **riduzione degli sprechi** (muda) e sul **miglioramento continuo** del flusso di valore, sulla base di 7 principi.

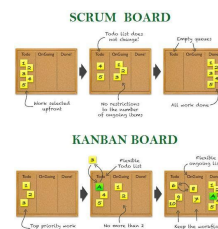


Fonte:
"Lean Software
Development:
An Agile
Toolkit" (2003) 15

Kanban



- E' una tecnica sviluppata in Giappone da **Toyota Motor Corporation** in ambito Operations Management
- Nell'ambito della gestione dei progetti, si basa sull'utilizzo di **strumenti visuali** (Kanban, board) per guidare il flusso di lavoro
- Le **carte Kanban** mostrano il lavoro programmato, il lavoro in progress e il lavoro completato
- Può essere integrato nell'approccio Scrum



Scrum



- In un articolo del 1986 Nonaka e Takeuchi (Harvard Business Review) paragonano il lavoro dei **team multi-funzionali** nei processi di sviluppo prodotti alle **formazioni scrum** dei team di rugby.
- Il termine viene ripreso negli anni '90 da Schwaber e Sutherland, poi dalla **Scrum Alliance**, per indicare un approccio agile, iterativo e collaborativo per la gestione di **progetti** di sviluppo software **complessi e innovativi**.



17



Scrum: principi e componenti chiave

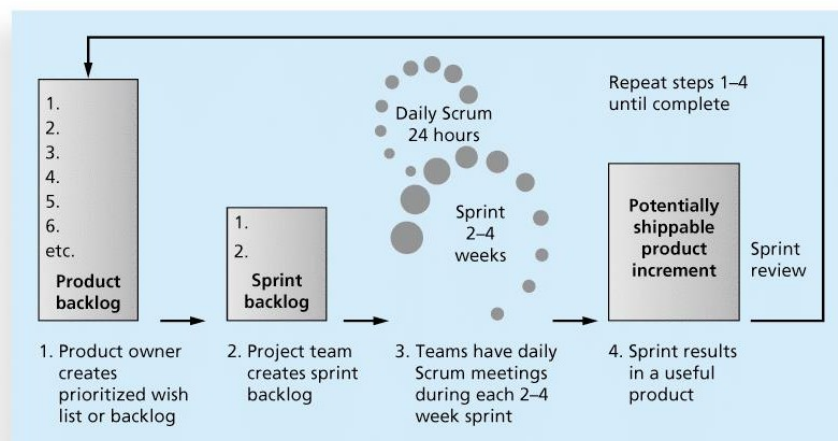
Caratteristiche dell'approccio Scrum



- Scrum è un approccio di **definizione del processo di sviluppo** che prevede di dividere il progetto in **blocchi rapidi di lavoro** (Sprint), in modo da creare alla fine di ciascuno di essi un incremento funzionante del prodotto (software).
- Il team è fortemente **responsabilizzato** nella definizione del lavoro da fare. L'approccio prevede **frequenti meeting di controllo e occasioni di feedback** da parte degli utilizzatori, per garantire adattabilità e flessibilità.
- **Elementi chiave dell'approccio Scrum:**
 - Ruoli
 - Tecniche (artefatti)
 - Eventi

19

L'approccio Scrum: una sintesi



Fonte: Schwalbe (2008) "Information Technology Project Management", Eighth Edition

20

I ruoli Scrum



- **Product owner:** la persona responsabile della massimizzazione del valore del prodotto. Ha la responsabilità finale sulla definizione del lavoro da svolgere (quale lavoro e in quale ordine), come documentato dal Product Backlog.
- **ScrumMaster:** la persona che facilita gli Scrum giornalieri, favorisce la stretta cooperazione tra tutti i ruoli e elimina le barriere per favorire il lavoro del team.
- **Scrum team:** Sono team multi-funzionali e auto-organizzati (in genere dalle 5 alle 9 persone). Decidono come organizzarsi internamente e come svolgere il lavoro, in modo da completare il lavoro desiderato ad ogni Sprint (che in genere dura dalle 2-4 settimane)

21

Le tecniche (artefatti) Scrum



- **Product backlog:** è un elenco ordinato di tutto ciò che potrebbe essere necessario al prodotto (features) ed è l'unica fonte di requisiti per le modifiche da apportare al prodotto.
- **Sprint backlog:** Le funzionalità a priorità più elevata del product backlog che devono essere completate entro uno sprint, più un piano di incremento del prodotto per raggiungere l'obiettivo dello Sprint.
- **Burndown chart:** mostra il lavoro cumulativo che rimane in un Sprint, su base giornaliera.

22

Product backlog e user stories

- Il **Product Backlog** è una lista ordinata (per priorità) di **elementi**, ognuno dei quali rappresenta un **pezzo di funzionalità** del sistema che si deve realizzare.
- Gli elementi seguono spesso il formato delle **User Stories**. Una User Story è una breve descrizione della funzionalità desiderata, vista dalla prospettiva dell'utilizzatore.
- I componenti del team dovrebbero poi associare una **stima dell'effort** (in ore, giorni, settimane, ecc...) richiesto per completare i compiti legati ad ogni user story.

23

Come si scrivono le User Stories

- **Format tipico** per scrivere una **userstory**:

"As a (role) I want (something) so that (benefit) "

"In qualità di (utente) vorrei (caratteristica) affinché (beneficio) "

Esempi di user stories

"As a frequent flyer, I want to rebook a past trip, so that I save time booking trips I take often"

"As a vice president of marketing, I want to select a holiday season to be used when reviewing the performance of past advertising campaigns so that I can identify profitable ones".

Linee guida su come scrivere User Stories

<https://www.mountaingoatsoftware.com/agile/user-stories#section-3>

<http://www.agilemodeling.com/artifacts/userStory.htm>

24

Dove scrivere User Stories?



Front of Card	Back of Card
<p>175</p> <p>As a student I want to purchase a parking pass so that I can drive to school</p> <p>Priority: High Should Estimate: 4</p>	<p>Comments:</p> <p>The student must pay the cost of the pass. One pass for one month is issued at a time. The student will not receive a pass if the payment isn't sufficient.</p> <p>The person buying the pass must be a currently enrolled student.</p> <p>The student may only buy one pass per month.</p>

Copyright 2005-2009 Scott W. Ambler

User Cards



Post-it

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	...
As a member, I can read profiles of other members so that I can find someone to date.	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 tests ...	4	4	1	0		
	Code the other ...	8	8	8	8		
As a member, I can update my billing information.	Update security tests	6	6	4	0		
	Design a solution to ...	12	6	0	0		
	Write test plan	8	8	4	0		
	Automate tests ...	12	12	10	6		
	Code the ...	8	8	8	4		

Software

25

Esempio di Product Backlog



ToDo List

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		30	

Esempio di Product Backlog per il sito web della Scrum Alliance:

<https://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/product-backlog/example>

26

Lo Sprint



- Il cuore di Scrum è uno **Sprint**. Ha una durata massima di un mese o meno durante il quale viene creato un **incremento di prodotto potenzialmente rilasciabile**, utilizzabile e “Fatto”.
- Gli Sprint hanno una **durata costante** durante il lavoro di sviluppo. Un nuovo Sprint si avvia immediatamente dopo la conclusione dello Sprint precedente.
- Ogni Sprint può essere considerato **un progetto** con un orizzonte non più lungo di un mese. Come i progetti, gli Sprint sono utilizzabili per realizzare qualcosa.

27

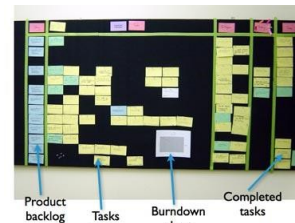
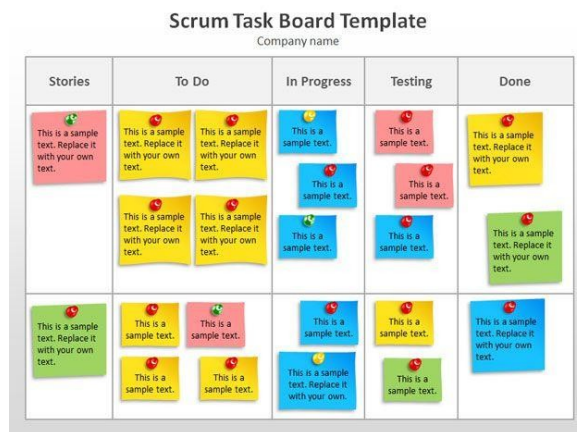
Gli eventi Scrum



- **Sprint planning session:** Un team meeting per selezionare, dal Product Backlog il lavoro da svolgere durante uno Sprint.
- **Daily Scrum:** Una riunione giornaliera rapida (15 minuti) del team di sviluppo, per condividere i progressi e i problemi e pianificare il lavoro del giorno successivo.
- **Sprint review:** un incontro nel quale il team mostra al cliente cosa ha completato durante lo Sprint.
- **Sprint retrospective:** un incontro in cui il team riflette su modalità per migliorare il prodotto e il processo, sulla base delle performance e delle lezioni apprese fino a quel momento

28

Strumenti visuali: Scrum Board



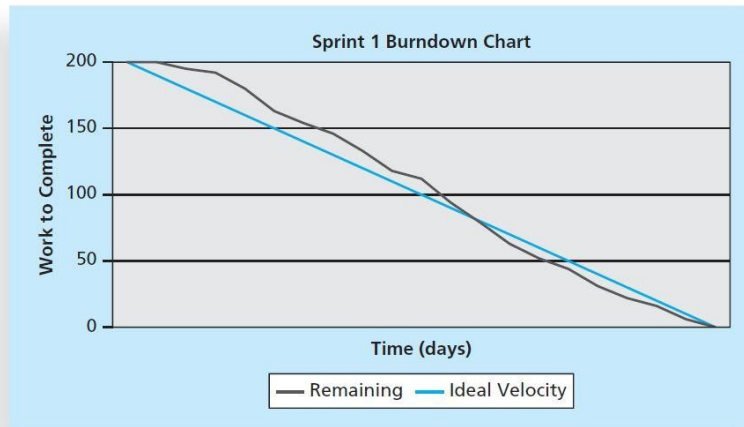
<https://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/task-boards>

SW per la gestione di sprint



www.trello.com

Esempio di Burndown Chart

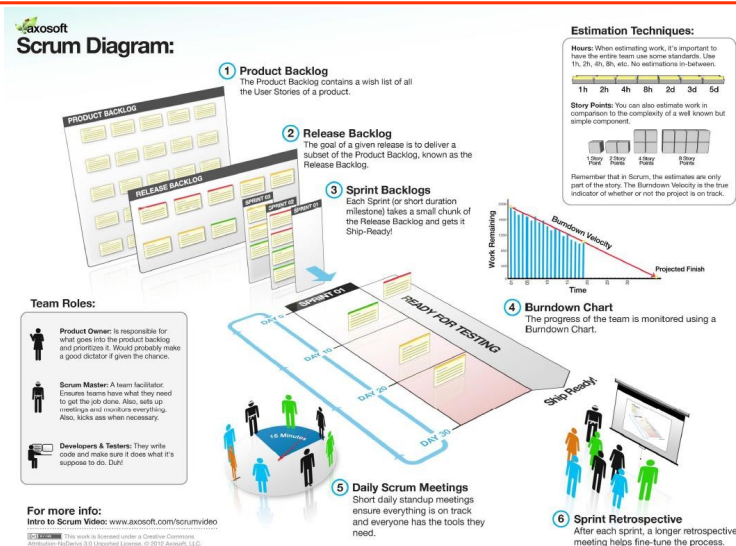


© Cengage Learning 2014

Information Technology Project Management, Eighth Edition

31

L'approccio Scrum: una sintesi



See also Axosoft Video «Scrum in 10 minutes» (<https://www.youtube.com/watch?v=XU0IRtyFM0>)

32

Adottare approcci Agili?



- Progetti con vincoli esterni meno rigidi, team con maggiore esperienza e preferibilmente co-localizzati, requisiti incerti, e maggiore flessibilità nei tempi possono essere **maggiormente adeguati** per questo approccio.
- I **principi** e le tecniche Scrum/Agile possono **essere adattati e modificati** a seconda delle caratteristiche dei progetti e del loro contesto.
- Uno dei problemi principali per imprese che sviluppano prodotti/software è rendere i **contratti compatibili** con approcci Scrum.
- L'adozione dell'approccio Agile richiede un forte **investimento iniziale in formazione** e condivisione dei valori chiave (esempio John Deere lettura HBR).

33

Le certificazioni per l'Agile PM



- Il Project Management Institute (PMI) ha riconosciuto l'interesse crescente negli approcci Agili introducendo una nuova certificazione nel 2011 chiamata "**Agile Certified Practitioner (ACP)**".
- Certificazione della Scrum Alliance per "**Certified Scrum Professional (CSP)**".



PMI Agile Certified Practitioner(PMI-ACP)®



34