



Processi software e sviluppo agile

I processi (di sviluppo del) software



- Un processo software descrive quali sono le attività che concorrono a sviluppare un prodotto software e come le attività sono collegate tra loro
- Assunzione: la qualità del processo implica la qualità del prodotto

Attività tipiche nello sviluppo ed evoluzione del software



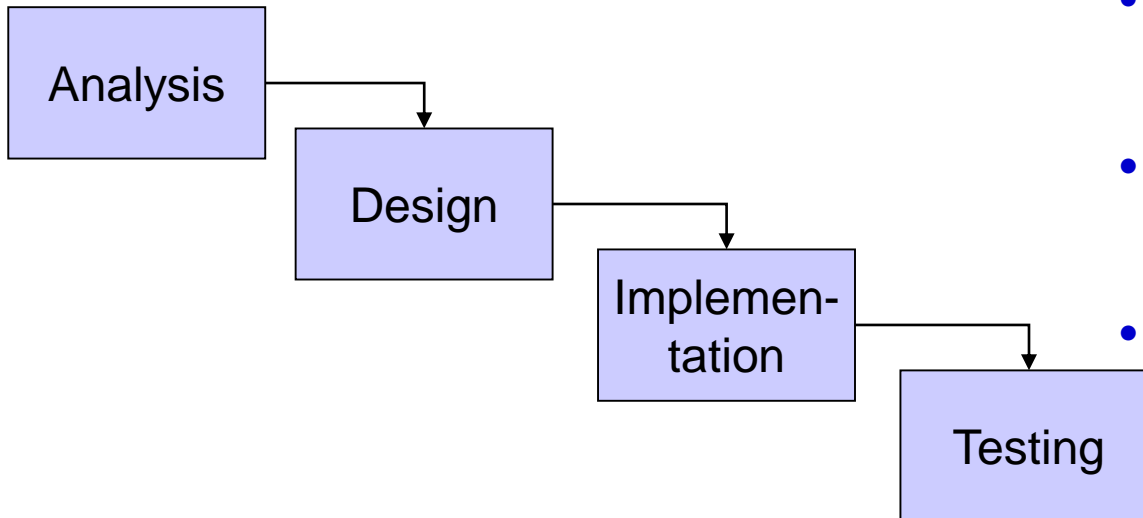
- Attività tecniche
 - Analisi dei requisiti (requirements analysis)
 - Progettazione (design)
 - Realizzazione (implementation)
 - Collaudo (testing)
 - Messa in esercizio (deployment)
 - Conduzione operativa (operation)
 - Manutenzione (maintenance)
- Attività organizzative
 - Gestione del progetto (project management)
 - Gestione della configurazione (configuration management)
 - Assicurazione della qualità (quality assurance)

Stile di processo a cascata (waterfall)



MINIMUM VIABLE PRODUCT: DEMO PER GLI
EARLY USERS

- Suddivide il progetto in base alle attività tecniche
 - Le fasi coincidono con le attività
- Si passa a una fase successiva solo se si completa l'attività e si supera il punto di controllo
 - Tornare indietro è possibile ma come eccezione



Problemi

- Rischio elevato: difficile stabilire che tutto procede veramente bene
- Difficile da applicare se i requisiti sono poco noti o volatili
- Time to market ritardato

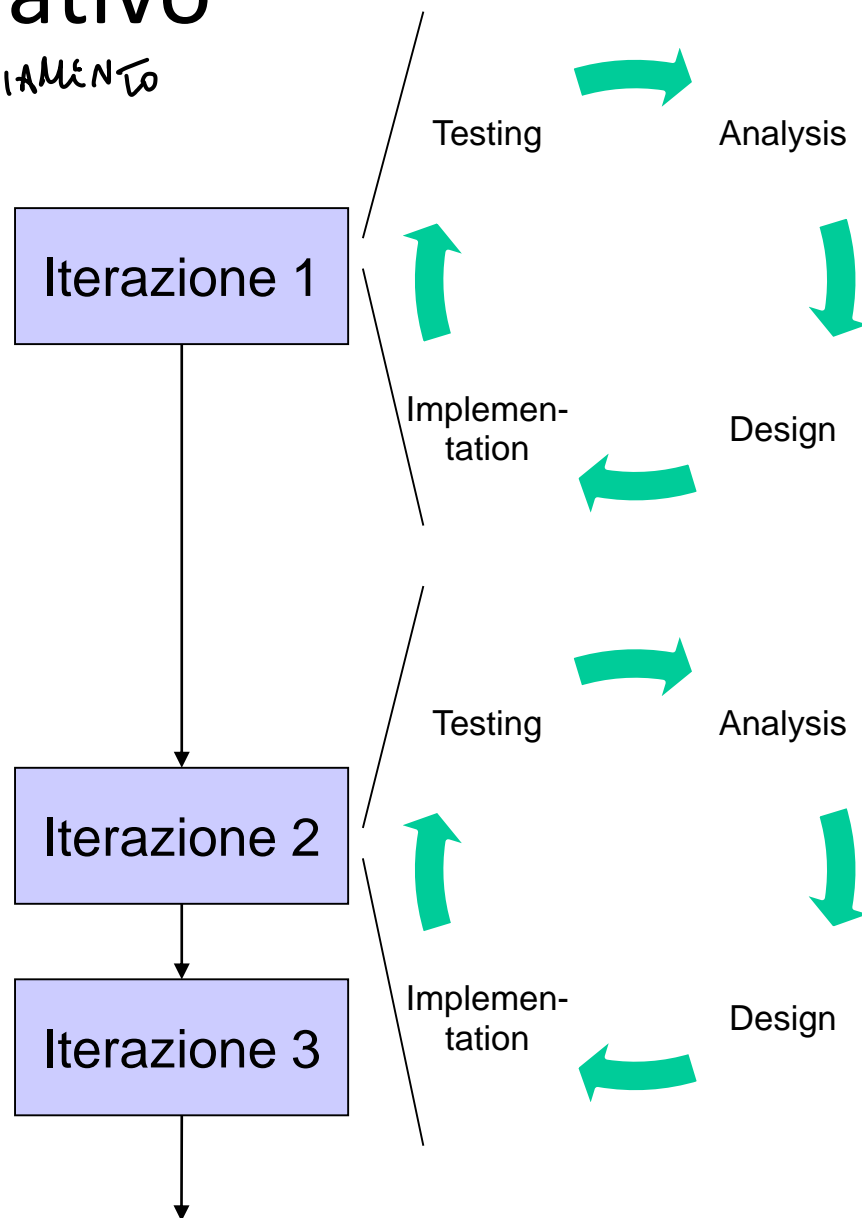


Stile di processo iterativo

È FLESSIBILE, SI ADATTA MEGLIO AL CAMBIAMENTO

- Anche conosciuto come incrementale, a spirale, evolutivo
- Suddivide il progetto in base a sottoinsiemi di funzionalità (iterazioni)
 - L'inizio delle iterazioni è preceduto da una fase esplorativa
 - Ogni iterazione produce codice (build) testato e integrato nel sistema complessivo
 - Le iterazioni messe in produzione sono dette release
 - Time boxing: le release sono rilasciate a intervalli di tempo regolari

PIÙ ITERAZIONI FACILITA A INTERVALLI
RIDOTTI, MENO RISCHIO COME E MENO INTARDO
GENERO



Non esiste un modo unico per strutturare lo sviluppo del software



TRANSIZIONE COMPLESSA
PER AZIENDE DI GRUPPO LANUBILE
COMPORTA DEI MSCI

- Sviluppo ad hoc (build and fix)
 - Conoscenza tacita
 - Documentazione inesistente o quasi
 - Approccio non ripetibile e trasferibile solo con apprendistato
- Sviluppo pianificato (plan-driven)
 - Piano dettagliato delle attività creato a monte
 - Molta documentazione come forma di comunicazione indiretta
 - Stile di processo a cascata
 - Adatto per progetti con requisiti noti a priori e stabili nel tempo
- Sviluppo agile
 - Fortemente adattivo rispetto ai cambiamenti in corso d'opera
 - Poca documentazione: enfasi sulla comunicazione diretta tra persone
 - Stile di processo iterativo: iterazioni corte e di durata costante (time-boxed)
 - Esempio: Scrum

Manifesto per lo Sviluppo Agile di Software

Stiamo scoprendo modi migliori di creare software,
sviluppendolo e aiutando gli altri a fare lo stesso.
Grazie a questa attività siamo arrivati a considerare importanti:

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

Il software funzionante più che la documentazione esaustiva

La collaborazione col cliente più che la negoziazione dei contratti

Rispondere al cambiamento più che seguire un piano

Ovvero, fermo restando il valore delle voci a destra,
consideriamo più importanti le voci a sinistra.

Kent Beck - Mike Beedle - Arie van Bennekum - Alistair Cockburn - Ward Cunningham - Martin Fowler -
James Grenning - Jim Highsmith - Andrew Hunt - Ron Jeffries - Jon Kern - Brian Marick - Robert C. Martin
- Steve Mellor - Ken Schwaber - Jeff Sutherland - Dave Thomas



I principi sottostanti al Manifesto Agile

Seguiamo questi principi:

I 12 principi sottostanti al Manifesto Agile



1. La nostra massima priorità è soddisfare il cliente rilasciando software di valore, fin da subito e in maniera continua.
2. Accogliamo i cambiamenti nei requisiti, anche a stadi avanzati dello sviluppo.
I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.
3. Consegniamo frequentemente software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.

I 12 principi sottostanti al Manifesto Agile



4. Committenti e sviluppatori devono lavorare insieme quotidianamente per tutta la durata del progetto.
5. Fondiamo i progetti su individui motivati. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.
6. Una conversazione faccia a faccia è il modo più efficiente e più efficace per comunicare con il team ed all'interno del team.

I 12 principi sottostanti al Manifesto Agile



7. Il software funzionante è il principale metro di misura di progresso.

8. I processi agili promuovono uno sviluppo sostenibile.

Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un ritmo costante.

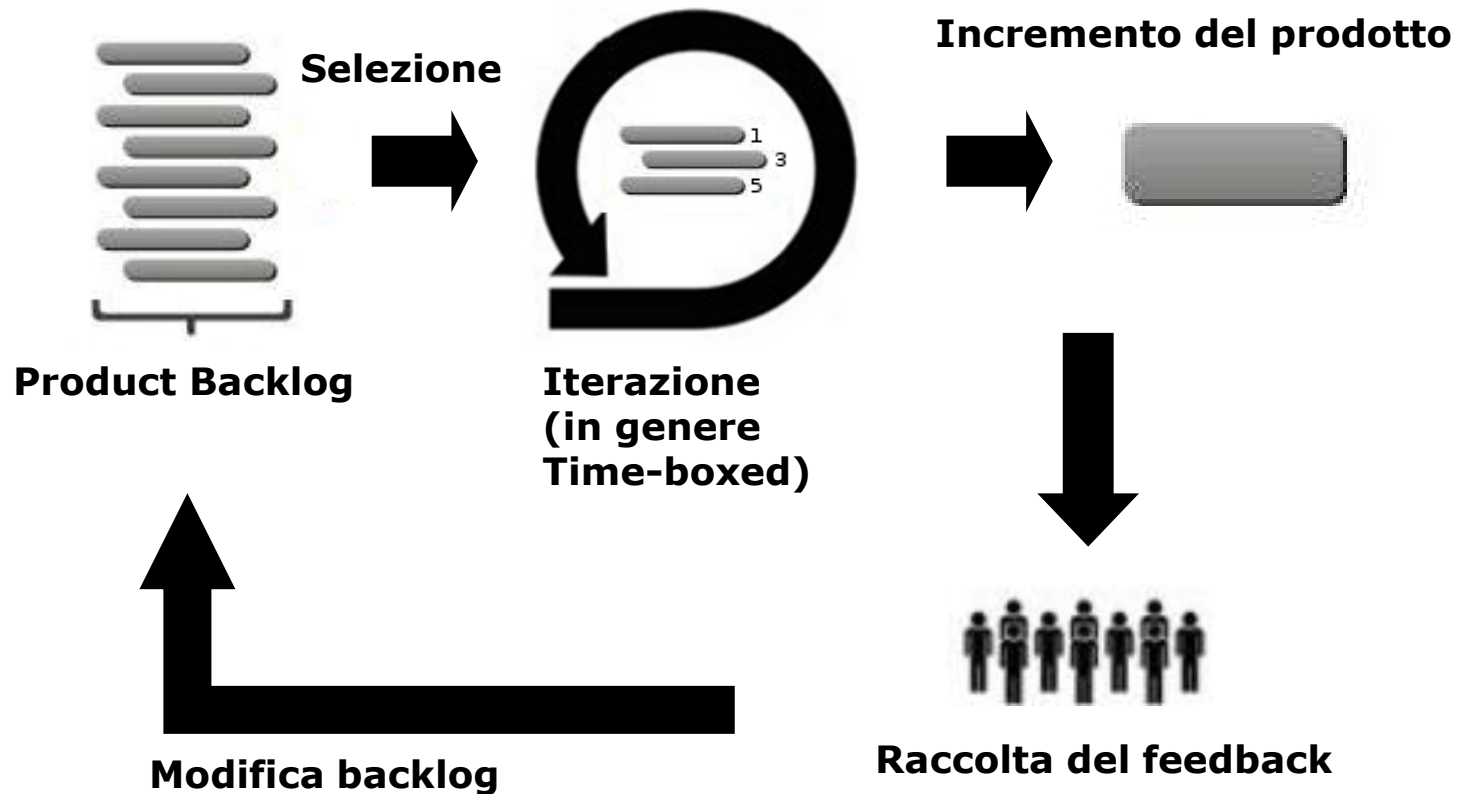
9. La continua attenzione all'eccellenza tecnica e alla buona progettazione esaltano l'agilità.

I 12 principi sottostanti al Manifesto Agile



- 10. La semplicità - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.
- 11. Le architetture, i requisiti e la progettazione migliori emergono da team che si auto-organizzano.
- 12. A intervalli regolari il team riflette su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

Un processo tipico di sviluppo agile del sw





Metodi agili

Piccoli medi progetti

- Extreme Programming (XP)
- Scrum
- Kanban

Grandi progetti

(> 50 persone o > 6 team)

- Scrum-of-Scrums
- Scaled Agile Framework (SAFe)
- Large-scale Scrum (LeSS)
- Disciplined Agile Delivery (DAD)
- Scaling Agile at Spotify (SA@S)