

Conclusioni del corso



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

Scopo del corso

Presentare e sperimentare *metodi e strumenti* di

- analisi,
- modellazione,
- progettazione, e
- misura

di sistemi e applicazioni software

Gli argomenti principali del modulo 1



Cosa abbiamo visto

- Gli standard di produzione del software
- Il ciclo di vita e il processo di sviluppo
- I metodi agili e Scrum
- L'analisi e la specifica dei requisiti
- La gestione dei progetti: tempi e costi
- Controllare e misurare la qualità del software

Esempi di domande-1

Quale delle seguenti affermazioni descrive meglio il compito principale del project manager di un prodotto software?

- A. Garantire che ogni sviluppatore scriva il codice meno costoso possibile.
- B. Gestire i requisiti parlando con gli stakeholder e analizzandoli col team di sviluppo.
- C. Gestire il testing in modo automatico cercando di risparmiare tempo e diminuire effort.
- D. Assicurarsi che il prodotto sia completato nei tempi previsti, rimanendo entro il budget e soddisfacendo i requisiti richiesti.

Esempi di domande-2

Qual è la principale differenza tra requisiti funzionali e requisiti non funzionali nel contesto dello sviluppo software?

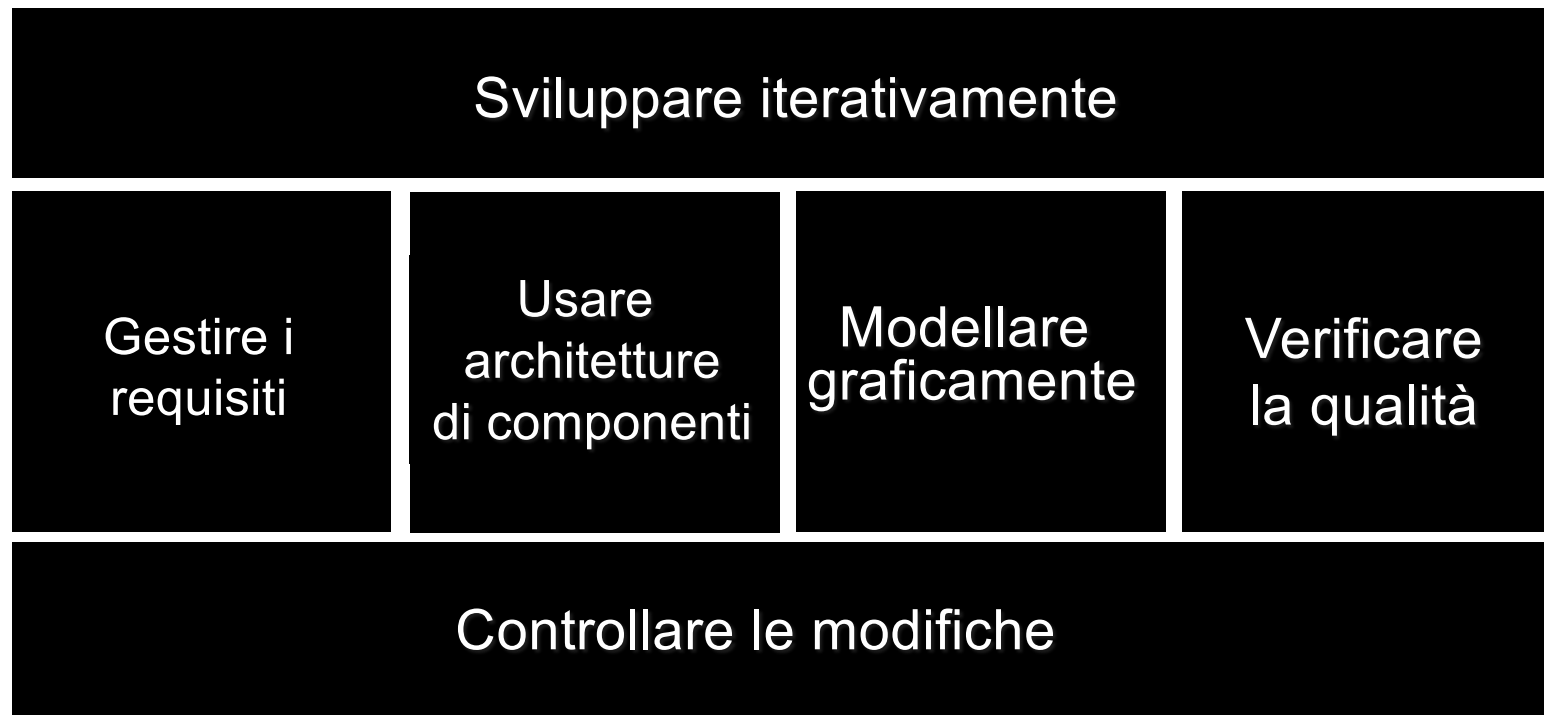
- A. I requisiti funzionali descrivono come il software deve essere implementato, mentre i requisiti non funzionali descrivono cosa il software deve fare.
- B. I requisiti funzionali sono sempre obbligatori, mentre i requisiti non funzionali sono opzionali
- C. I requisiti funzionali definiscono i comportamenti e le operazioni del sistema, mentre i requisiti non funzionali specificano vincoli o caratteristiche di qualità del sistema.
- D. I requisiti funzionali riguardano solo l'interfaccia utente, mentre i requisiti non funzionali riguardano solo il backend del sistema.

Esempi di domande- 3

Quale delle seguenti caratteristiche identifica meglio un processo di sviluppo agile?

- A. La suddivisione del lavoro in iterazioni brevi e incrementali con feedback frequenti dagli stakeholder.
- B. L'uso delle user story per definire completamente i requisiti di prodotto prima di iniziare lo sviluppo.
- C. La pianificazione precisa di tutte le attività necessarie al prodotto prima dell'avvio dello sviluppo.
- D. L'assegnazione di un singolo responsabile che prenda tutte le decisioni tecniche e di progetto.

Principi guida dello sviluppo software



“Leggi” dello sviluppo del software

1. Legge di Joel: Scrivere software è molto più facile che leggerlo
2. [Legge di Brooks](#): aggiungere personale per rimediare ai ritardi di un progetto in ritardo lo fa ritardare di più
3. Il principio di incertezza (legge di Ziv): lo sviluppo di un software è imprevedibile; lo sforzo necessario allo sviluppo può risultare al consuntivo da 4 volte a $\frac{1}{4}$ della stima iniziale preventiva
4. Legge di Humphrey: i requisiti di un nuovo sistema non saranno chiari finché gli utenti non iniziano a usarlo, ovvero un problema non è ben compreso finché non si sviluppa una soluzione
5. Lemma di Wegner: non è possibile specificare o testare completamente un sistema interattivo (perché avete bisogno di vedere come lo usano gli utenti)
6. Le leggi di Lehman sull'evoluzione del software: i programmi grandi diventano parte del mondo reale, lo cambiano ed evolvono insieme ad esso

1. [C. Jones, A Retrospective View of the Laws of Software Engineering, Crosstalk, 2017](#)
2. [The laws of software development](#)

Il software è un costrutto sociale

- Sviluppare da soli?
- Sviluppare in due?
- Sviluppare in team?
- Sviluppare in tanti?
- Sviluppare con AI?

Stiamo scoprendo modi migliori di costruire il software facendolo e **aiutando altri a farlo**. Attribuiamo valore a

Individui e interazioni

Software che funziona

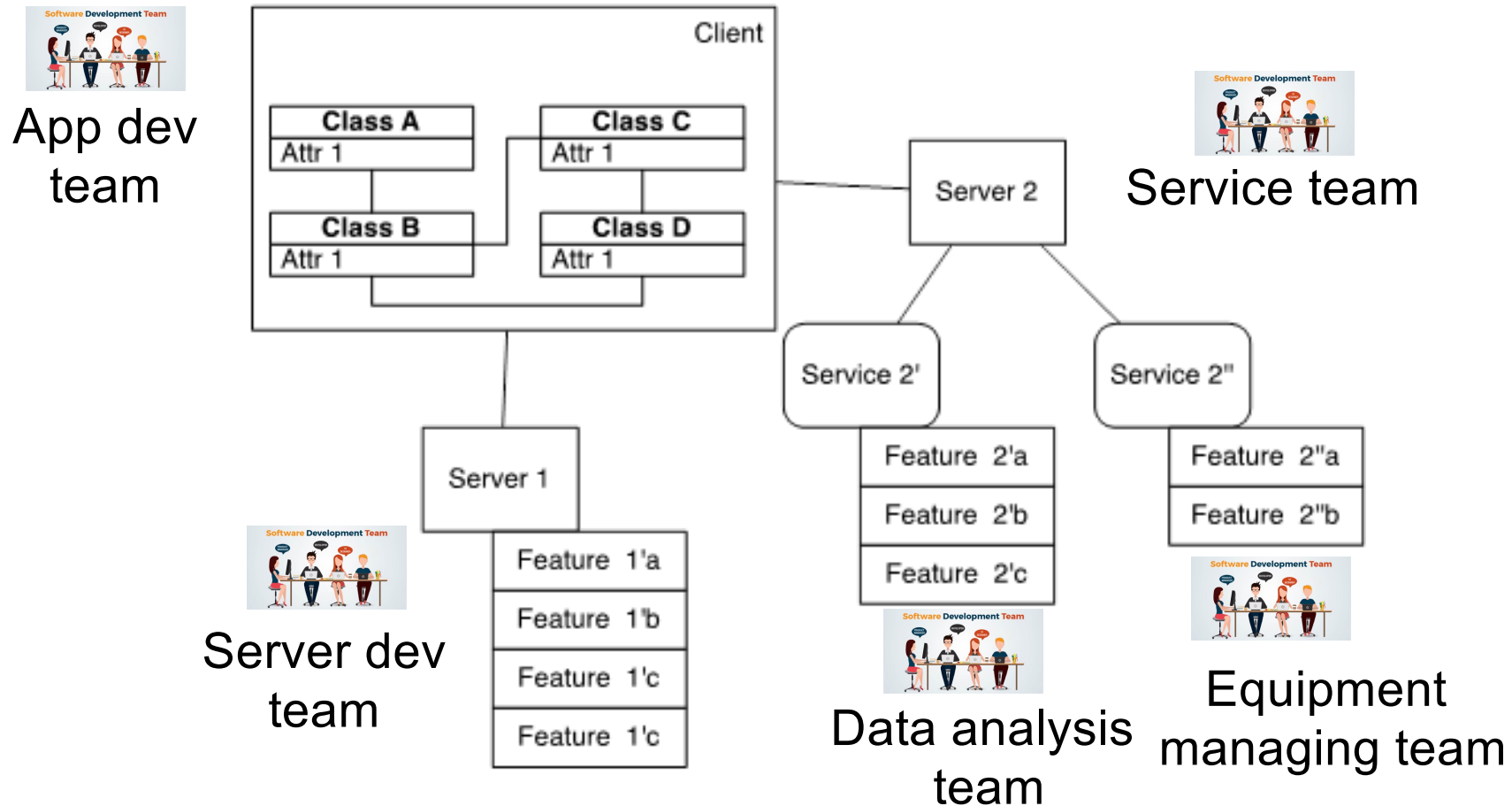
Collaborazione col cliente

Reagire al cambiamento

Manifesto agile

Legge di Conway:

L'architettura di un sistema software riflette la struttura dell'organizzazione che costruisce un sistema

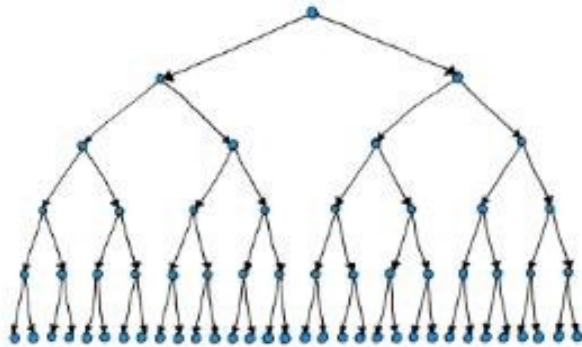


Legge di Conway

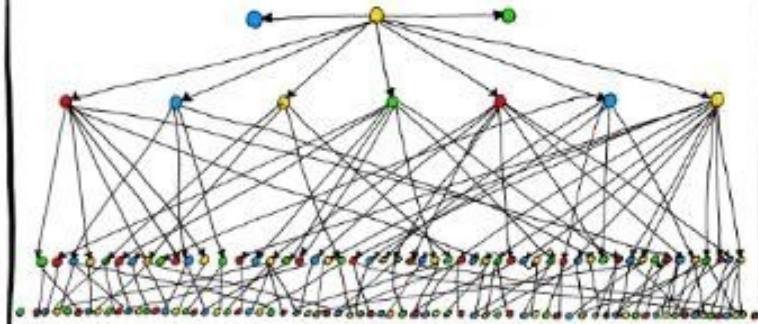
(versione di Paolo)

Il software è il prodotto di un
processo sociale,
e ne incorpora la struttura

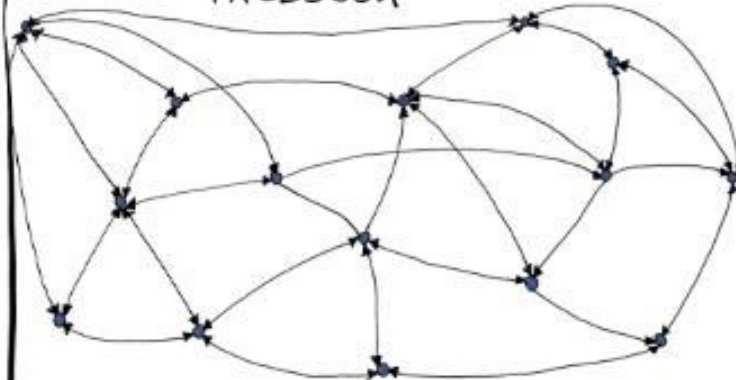
AMAZON



GOOGLE



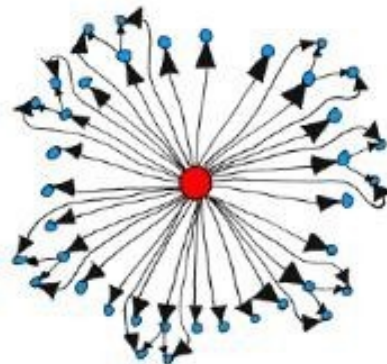
FACEBOOK



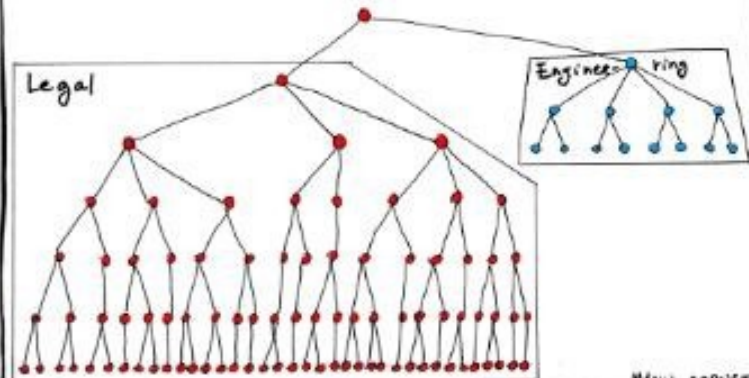
MICROSOFT



APPLE



ORACLE



Adesso sapete rispondere?

- Come si *modella* il software?
- Come si *sviluppa* il software?
- Come si *riusa* il software?
- Quali *strumenti* sono disponibili per chi costruisce software?
- Quanto *costa* costruire il software?
- Quanto *tempo* ci vuole?
- Come si valuta la *qualità* del software?

Proposte di tesi di laurea

- Automazione del testing software attraverso l'uso di tecniche di AI
- Uso di LLM come supporto al lavoro del PO
- Chatbot basato su Essence come supporto al lavoro dello SM
- Uso di CoPilot in sviluppi agili con Extreme Programming
- Metodi di sviluppo di software per giochi di strategia

*Le persone che amano le salsicce e il
software non dovrebbero mai chiedersi
come vengono fatti né le une né l'altro*

David Lee Todd, Product Manager

Domande?

