

Conclusioni del corso



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

*People who love sausages and software
should never watch either being made*

David Lee Todd, Product Manager

Scopo del corso

Presentare e sperimentare *metodi e strumenti* di

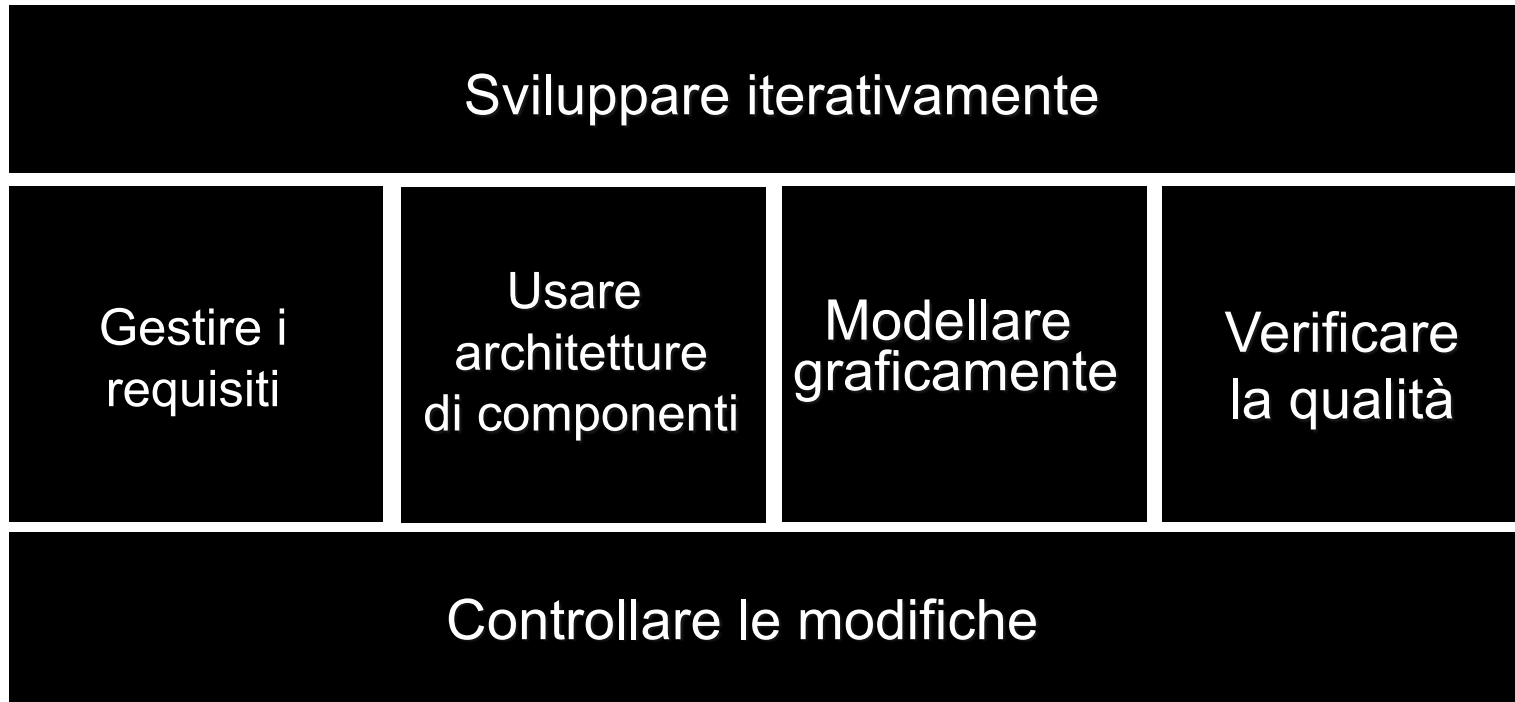
- analisi,
- modellazione,
- progettazione, e
- misura

di sistemi e applicazioni software

Cosa abbiamo visto

- Gli standard di produzione del software
- Il ciclo di vita dei prodotti software
- La modellazione del software con UML
- L'analisi e la specifica dei requisiti
- La progettazione del software
- Project Management per sistemi software
- Controllare e misurare la qualità del software
- La manutenzione dei sistemi software
- Strumenti di gestione delle configurazioni

Principi guida dello sviluppo software



“Leggi” dello sviluppo del software

1. Legge di Joel: Scrivere software è molto più facile che leggerlo
2. Il principio di incertezza (legge di Ziv): lo sviluppo di un software è imprevedibile; lo sforzo necessario allo sviluppo può risultare alla fine da 4 volte a $\frac{1}{4}$ della stima iniziale
3. 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
4. Lemma di Wegner: non è possibile specificare o testare completamente un sistema interattivo
5. Legge di Brooks: aggiungere personale per rimediare ai ritardi di un progetto in ritardo lo fa ritardare di più
6. Le leggi di Lehman sull'evoluzione del software: i programmi grandi diventano parte del mondo reale, lo cambiano ed evolvono insieme ad esso

- C. Jones, A Retrospective View of the Laws of Software Engineering, Crosstalk, 2017
- <http://www.eric-stewart.com/blog/laws-of-software-development/>

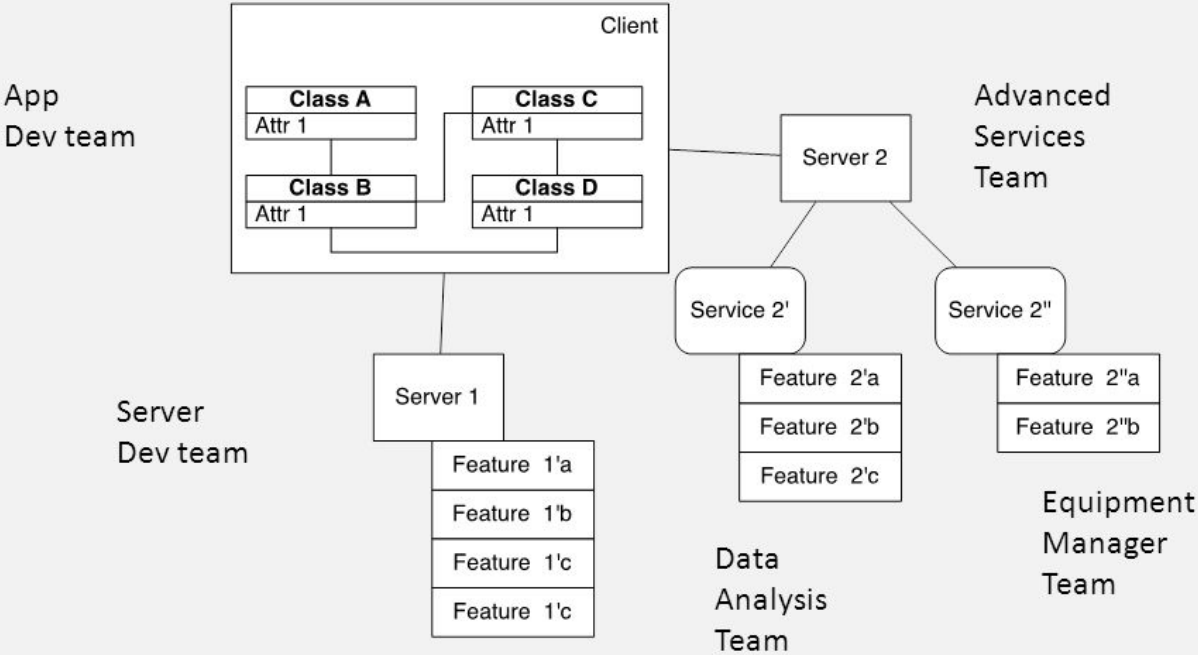
Legge di Conway

(versione di Paolo)

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

Conway's Law

The architecture will reflect the structure of the organization that built the system!



Le sfide

- Fare software è difficile, i metodi e gli strumenti dell'ingegneria del software aiutano
- Anticipare i trend futuri: partecipare alla ricerca e incorporare i risultati nella disciplina
- Separare i principi eterni dalle pratiche superate
- Aggiornare continuamente il materiale didattico
- Conoscere le pratiche industriali e applicarle in progetti didattici anche piccoli
- Aiutare i neofiti a imparare a imparare
- Offrire aggiornamento ai professionisti

Adesso sapete rispondere?

- Come si *modella* il software?
- Come si *progetta* 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?

Domande?

