

# Introduzione al progetto

---

Fabio Vitali  
22 ottobre 1999



# Introduzione

Oggi esaminiamo in breve:

- ◆ L'architettura del macro-progetto
- ◆ La divisione in temi
- ◆ La divisione in famiglie



# Il macro-progetto

- Un unico grande progetto a cui partecipano tutti i frequentanti
- Enfasi sulla collaborazione trasversale e continua (come negli organismi di standard)
- Non ci sono gruppi:
  - ◆ Famiglie (accomunate dall'usare gli stessi strumenti per temi diversi)
  - ◆ Temi (accomunate dal realizzare lo stesso programma con strumenti diversi)



# Il problema semplice



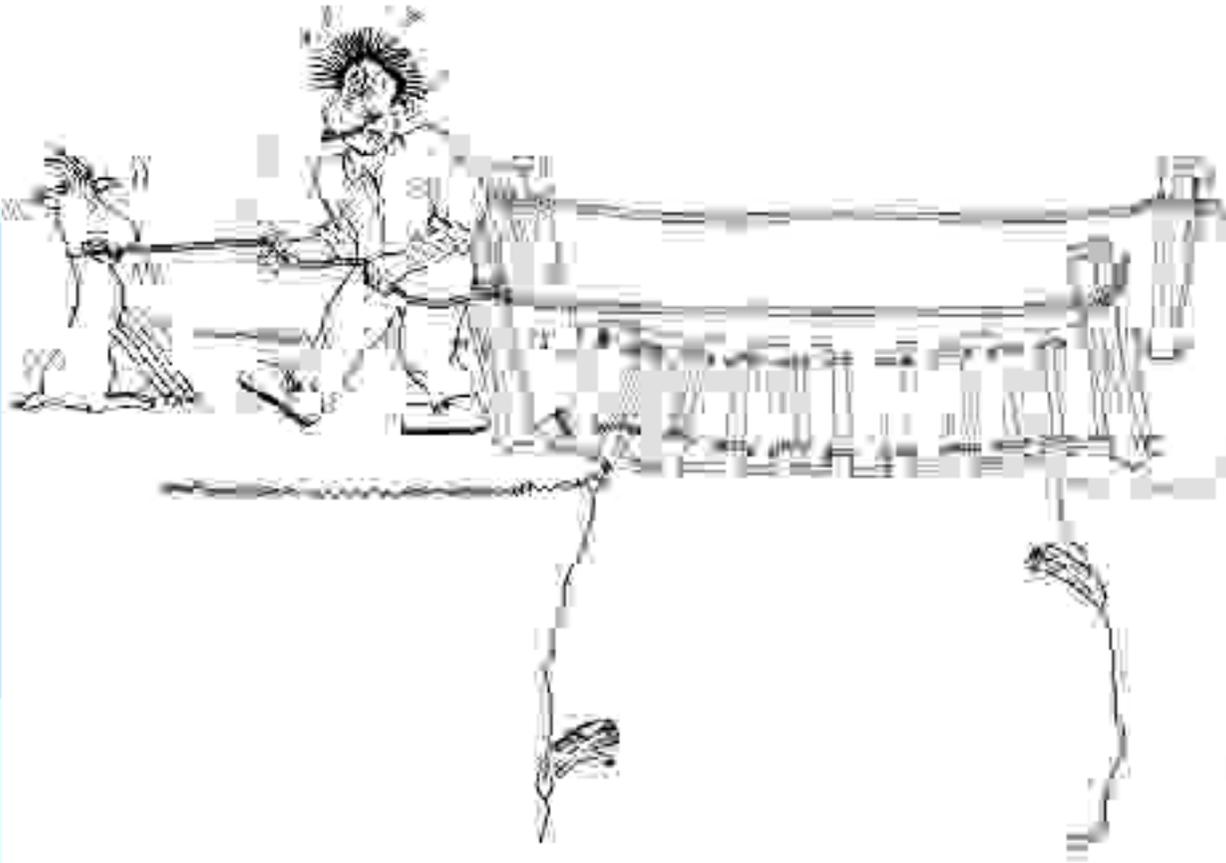
# Il problema semplice risolto



# Il problema difficile



# Il problema difficile risolto



# Le famiglie (1)

Tutti i frequentanti si dividono in tre famiglie:

- ◆ MICROSOFT: strumenti COM Microsoft: Visual Basic, Visual C++, ecc.
- ◆ SUN: una qualunque versione *standard* di Java (ad esempio, Visual J++ non va bene)
- ◆ GNU: strumenti e linguaggi disponibili nelle distribuzioni GNU: C, C++, Perl, Python, TCL/TK, emacs, ecc.



# Le famiglie (2)

La distribuzione in famiglie è volontaria

Tuttavia, le famiglie debbono essere di composizione omogenea per numero e qualità dei programmatori.

Lo scopo ultimo della famiglia è creare un software funzionante e complesso dalla somma delle attività dei singoli.

Il nome del gioco è **integrazione.**



# I temi (1)

Il progetto riguarda la realizzazione di un sistema molto complesso

Il sistema viene quindi diviso in tanti sottoproblemi, detti *temi*, attraverso discussione comune.

Tutte le famiglie debbono realizzare un'implementazione interoperabile di ciascun tema identificato

Le implementazioni dei temi debbono, riunite in un unico sistema, fornire tutte le funzionalità richieste



# I temi (2)

Bisogna ipotizzare di far parte di un gruppo di standardizzazione (tipo IETF o W3C)

Ciascun individuo parla a nome della famiglia cui appartiene

E' necessario trovare i compromessi più adatti per favorire la propria implementazione senza però imporsi agli altri

La ricerca del consenso è fondamentale



# I temi (3)

Il progetto riguarderà e richiederà l'uso di tecnologie tipiche del World Wide Web.

E' obbligatorio che le implementazioni usino o estendano protocolli e tecniche esistenti nel World Wide Web.

Ogni deviazione da standard esistenti deve essere giustificata e approvata da tutti i membri di un tema. In più mi dovete convincere (N.B.: non è facile).



# I temi (4)

La scelta dei temi è libera

Tuttavia, i temi debbono essere scelti omogeneamente tra le famiglie, e riflettere scelte architettoniche importanti

Lo scopo ultimo del tema è trovare un protocollo ragionevole che permetta ad implementazioni diverse di svolgere compiti analoghi e di comunicare tra loro.

Il nome del gioco è **interoperabilità**.



# Architettura (1)

Ogni famiglia può strutturare l'architettura interna come meglio preferisce

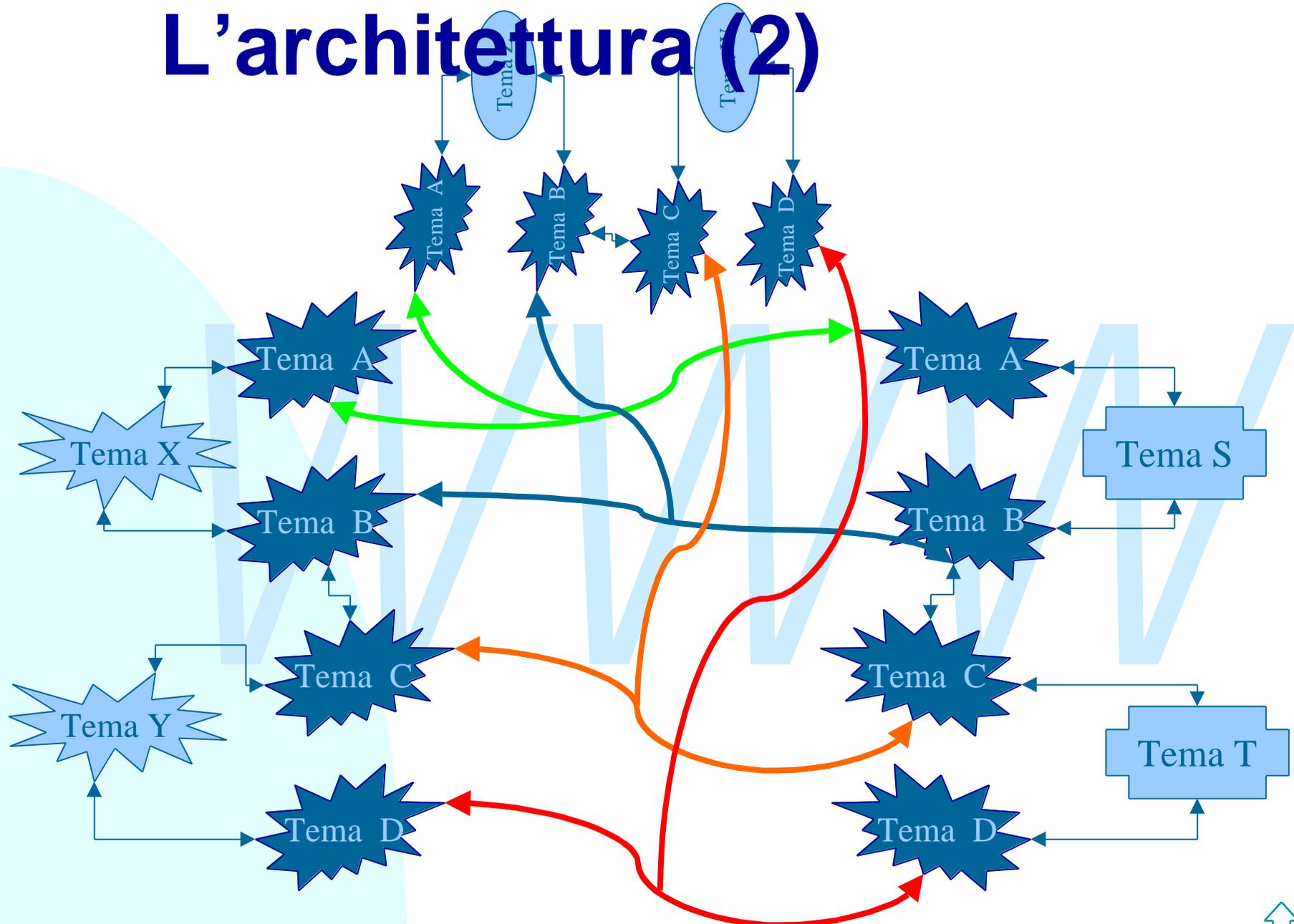
Tutti i ruoli "pubblici" identificati dalla discussione comune debbono essere coperti

Tutti i ruoli "privati" identificati dalle famiglie debbono essere coperti

La discussione deve stabilire le interfacce tra i temi: protocolli tra famiglie diverse, API tra temi della stessa famiglia



# L'architettura (2)



# Ruoli extra

Esistono tre ruoli da coprire all'interno di ogni famiglia:

- ◆ Il “guru” possiede profonde conoscenze tecniche sugli strumenti prescelti e consiglia gli altri.
- ◆ Il “project manager” stabilisce fasi, deadline e criteri di confronto e organizzazione del progetto, ed è responsabile della corretta realizzazione del sistema
- ◆ Il “system administrator” installa e mette a disposizione strumenti e reti. Per esempio installa il server di news da utilizzare per le discussioni

Questi ruoli possono essere coperti da persone già impegnate in temi specifici (vale più punti) oppure possono esserne l'unico compito

I ruoli extra vanno coperti per acclamazione all'interno delle famiglie



# Perché tutto questo?

- Si vuole simulare quanto accade nei gruppi di standard: discussioni, ricerca del consenso, generazione del protocollo, verifica e modifica
- Il processo di realizzazione deve avvenire in tre fasi:
  - ◆ Discussione sul problema (nasce il tema)
  - ◆ Ricerca del consenso (nasce il protocollo)
  - ◆ Implementazione (viene testata l'interoperabilità)



# Il processo

Specifichiamo meglio il processo di realizzazione del sistema:

- ◆ I fase: da adesso a Natale '99: divisione in famiglie, identificazione dei temi, attribuzione dei temi ai singoli, creazione delle architetture di discussione.
- ◆ II fase: da gennaio a Pasqua 2000: Tema per tema, identificazione delle dipendenze reciproche e creazione di API e protocolli.
- ◆ III fase: da Pasqua a giugno 2000: implementazione, test, integrazione e seconda versione del protocollo.



# I fase (fino a Natale '99)

- A breve riceverete il documento dei requirement (il problema).
- Tramite discussione globale (usando il newsgroup di IUM) vi dividete in famiglie.
- Le famiglie identificano al loro interno le persone che copriranno i ruoli speciali (guru, project manager, system administrator)
- I temi proposti verranno analizzati, tramite discussione globale, e rivisti, spezzati, riorganizzati.
- Le famiglie individuano, tema per tema, la/le persone a cui affidare il tema.



# Il fase (fino a Pasqua 2000)

- Vengono creati i newsgroup (o le mailing list) di famiglia e di tema. Ciascuno è tenuto a partecipare alle discussioni (fanno parte integrante del voto finale)
- In questa fase vengono realizzate le specifiche di sistema ed i protocolli di interoperabilità.
- Le specifiche di sistema vengono realizzate attraverso discussione intra-familiare secondo criteri illustrati a Ingegneria del software
- I protocolli di interoperabilità vengono realizzati attraverso discussione inter-familiari, tema per tema, secondo criteri illustrati a IUM.
- E' lecito ma non garantito iniziare la fase di implementazione prima della conclusione dei protocolli.



## III fase (fino a giugno 2000)

- Vengono implementati i moduli descritti dai documenti della II fase.
- I moduli vengono testati separatamente, integrati e testati insieme. Vengono anche eseguiti i test di interoperabilità.
- I tester compilano bug report che vengono ricevuti, catalogati e affrontati in ordine di importanza.
- Il progetto si conclude in un punto a scelta tra il livello minimo ed ottimale di correttezza del sistema integrato, identificati in precedenza.
- L'esperienza evidenzia problemi di interoperabilità che portano ad una seconda versione dei protocolli.



# I protocolli (1)

- Ogni working group di tema deve elaborare tre o più documenti di protocollo:
  - ◆ *Obiettivi del working group*: limiti e funzionalità trattate, relazioni con gli altri working group
  - ◆ *Regole di interoperabilità*: processo, sintassi, gestione degli errori
  - ◆ *Criteri di verifica*: criteri minimi di interoperabilità (*core set*) e funzionalità opzionali (*optional set*). *Core test suite* e *optional test suite*.
- I documenti di protocollo debbono essere
  - ◆ sufficientemente completi da esprimere e lasciar implementare tutte le funzionalità richieste al tema
  - ◆ sufficientemente precisi da garantire l'effettiva interoperabilità delle implementazioni



# I lavori di discussione (2)

- La discussione avviene necessariamente e completamente su newsgroup. Riunioni fisiche sono possibili, ma i risultati vanno verbalizzati e esposti su newsgroup
- Scopo del working group è generare documenti di protocollo consensuali.
- Viene eletto all'inizio un *chair* del working group. Ha il compito di generare i documenti di protocollo; è il portavoce del tema con i docenti.
- Il chair deve ottenere il consenso a tutti i costi, trovando le situazioni di compromesso più opportune. Il protocollo viene sottoposto all'approvazione globale alla fine della II fase, e corretto adeguatamente.



# Il project manager

- Il P.M. decide l'organizzazione interna del progetto per famiglie. E' il chair del newsgroup della famiglia. E' anche il portavoce con i docenti. E' a carico del P.M.:
  - ◆ La composizione dei working group
  - ◆ L'identificazione delle fasi interne di progetto
  - ◆ I criteri di accettabilità del progetto (livello minimo ed ottimale)
  - ◆ La verifica del lavoro dei vari membri dei working group.
- Il P.M. ha il potere di spostare persone da un tema ad un altro, creare nuovi temi, stabilire criteri di eccellenza.
- Ci può essere da un minimo di uno ad un massimo di tre P.M. in famiglia, ma ogni decisione deve essere consensuale.



# Il system administrator

- Il system administrator deve fornire ai membri della famiglia gli strumenti per realizzare il loro lavoro:
  - ◆ Newsgroup di famiglia e di tema
  - ◆ Siti web per i documenti di lavoro e ogni altra informazione
  - ◆ Ambienti di programmazione e test
- Gestisce gli account ed il sito Web di famiglia
- C'è un solo system administrator per famiglia.



# Il guru

- Il compito del guru è fornire assistenza tecnica agli implementatori della sua famiglia.
- Esso fornisce consigli (half guru) o risolve i problemi (full guru) degli implementatori.
- I problemi propostigli possono solo essere relativi all'architettura di famiglia, non ai dettagli dell'implementazione.
- C'è al massimo un guru in ogni famiglia, e la famiglia decide se è half o full.



# L'implementatore

Lucido non  
mostrato a  
lezione

- L'implementatore ha sotto la sua responsabilità il compito di realizzare e testare uno o più moduli del sistema.
- La famiglia identificherà i compiti implementativi da affidare.
- Ad un *implementatore capo* verranno affidati incarichi di peso maggiore che ad un *implementatore semplice*, nell'ottica di una equa suddivisione dei compiti.
- Un implementatore capo non ha nessuna autorità su un implementatore semplice, la differenza di nome è unicamente connesso con il carico di lavoro affidato.



# Il tester

Lucido non  
mostrato a  
lezione

## I sistemi vengono testati in tre maniere

- ◆ Test del modulo: verifica della corretta implementazione del singolo modulo, fatta da membri della stessa famiglia ma non dall'implementatore.
- ◆ Test del sistema: verifica della corretta integrazione del sistema, fatta da membri di altre famiglie.
- ◆ Test dell'interoperabilità: verifica della corretta interoperabilità del sistema fatta da membri di altri working group.

In ogni caso MAI un implementatore verifica il proprio modulo.



# Le decisioni

Lucido non  
mostrato a  
lezione

- Le decisioni all'interno di qualunque contesto vengono prese democraticamente come risultato di discussioni. Vengono imposte dall'autorità costituita sono in caso di chiara mancanza di accordo.
- I contesti sono tre:
  - ◆ Il tema: decisioni semplici a maggioranza. Decisioni chiave a maggioranza qualificata (75% +1)
  - ◆ La famiglia: decisioni semplici a maggioranza. Decisioni chiave a maggioranza qualificata (75%+1)
  - ◆ La classe: decisioni semplici a maggioranza e mio assenso. Decisioni chiave a maggioranza qualificata (75%+1+il mio assenso) indipendentemente dall'appartenenza ad una famiglia o ad un working group.



# I ruoli e i punti

- Ogni ruolo ha un peso nel progetto, in fatica e responsabilità.
- Qui si dà un primo punteggio ai ruoli, che è soggetto in qualunque momento a verifica e cambiamento (ma si richiede il consenso mio e del 50% +1 della classe).
- Ogni studente di due annualità (IUM e IdS) deve prendersi ruoli per 16-24 punti. Ogni studente di una annualità deve prendersi ruoli per 8-12 punti.
- Studenti non di informatica non possono prendere ruoli di implementatore, guru, o system administrator.



# Una prima tabella di punti

Ruolo	Punti
project manager	8
half guru	4
full guru	8
system administrator	8
Implementatore semplice	8
Implementatore capo	16
Membro di working group	4
Chair di working group (in aggiunta a membro)	4
Tester di modulo	2
Tester di sistema	2
Tester di interoperabilità	2



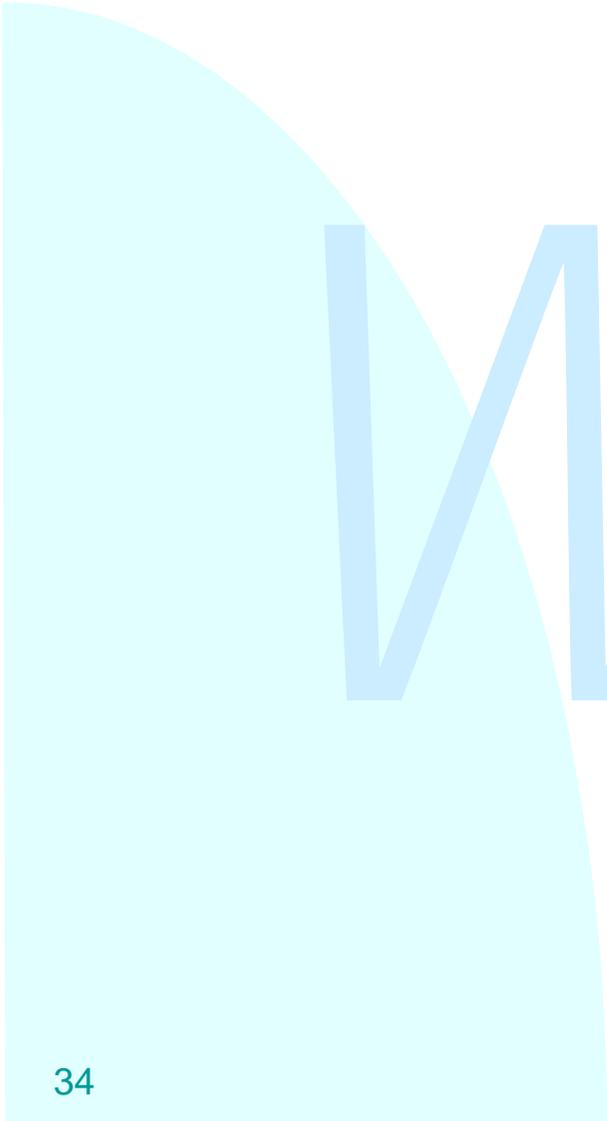
# Il progetto e il corso (a)





```
if (...) {...} else {...}  
while (;;) {...}  
    i++
```





www



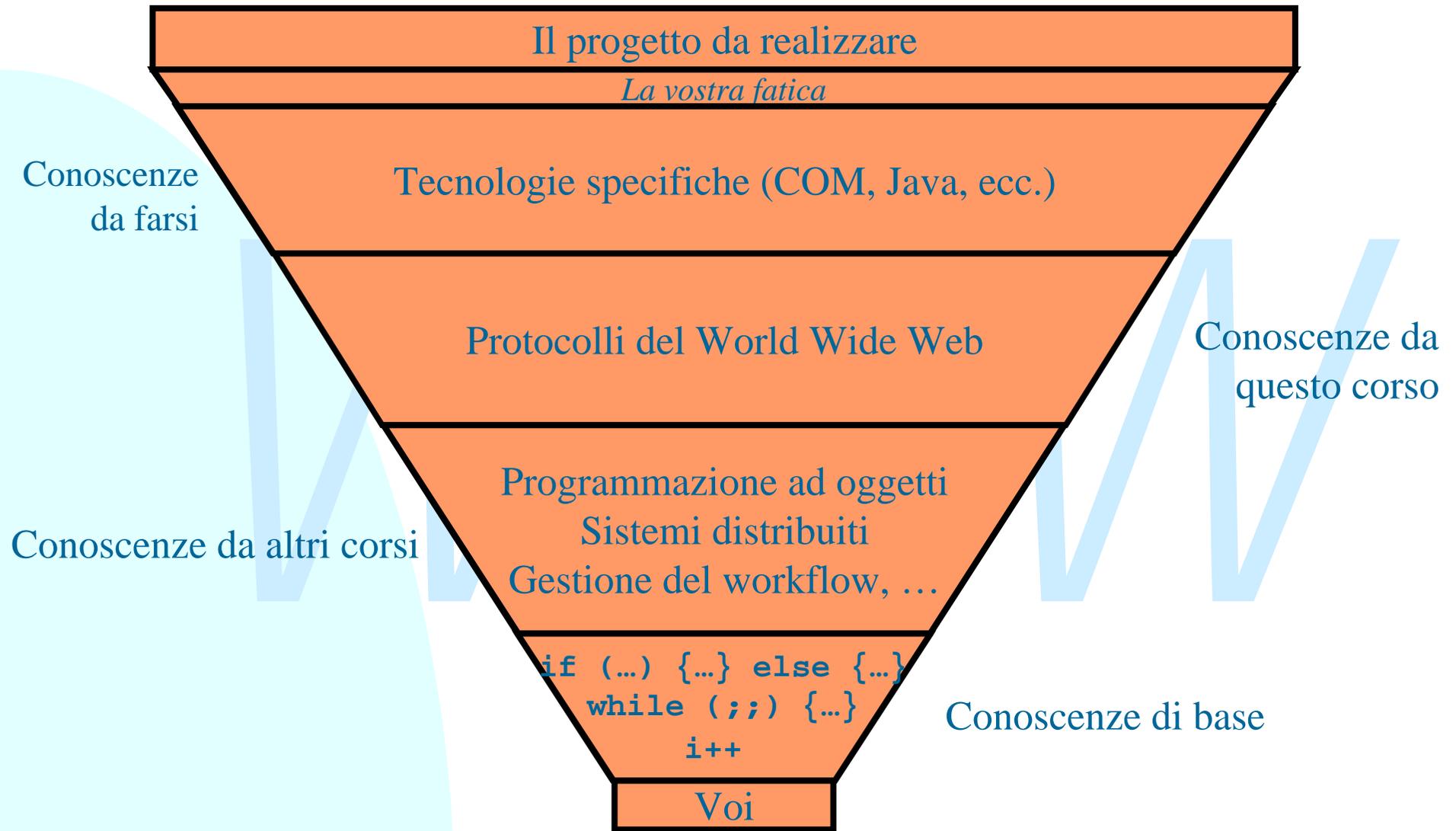
# Il progetto e il corso (d)



# Il progetto e il corso (e)



# Il progetto e il corso (f)



# Conclusioni

Il progetto vuole dare un'idea del lavoro d'equipe e del funzionamento dei gruppi di standard

La divisione in famiglie permette di accedere nei dettagli ad una di tre tecnologie interessanti

La divisione in temi permette di realizzare progetti importanti e senza ridondanze e ripetizioni

