

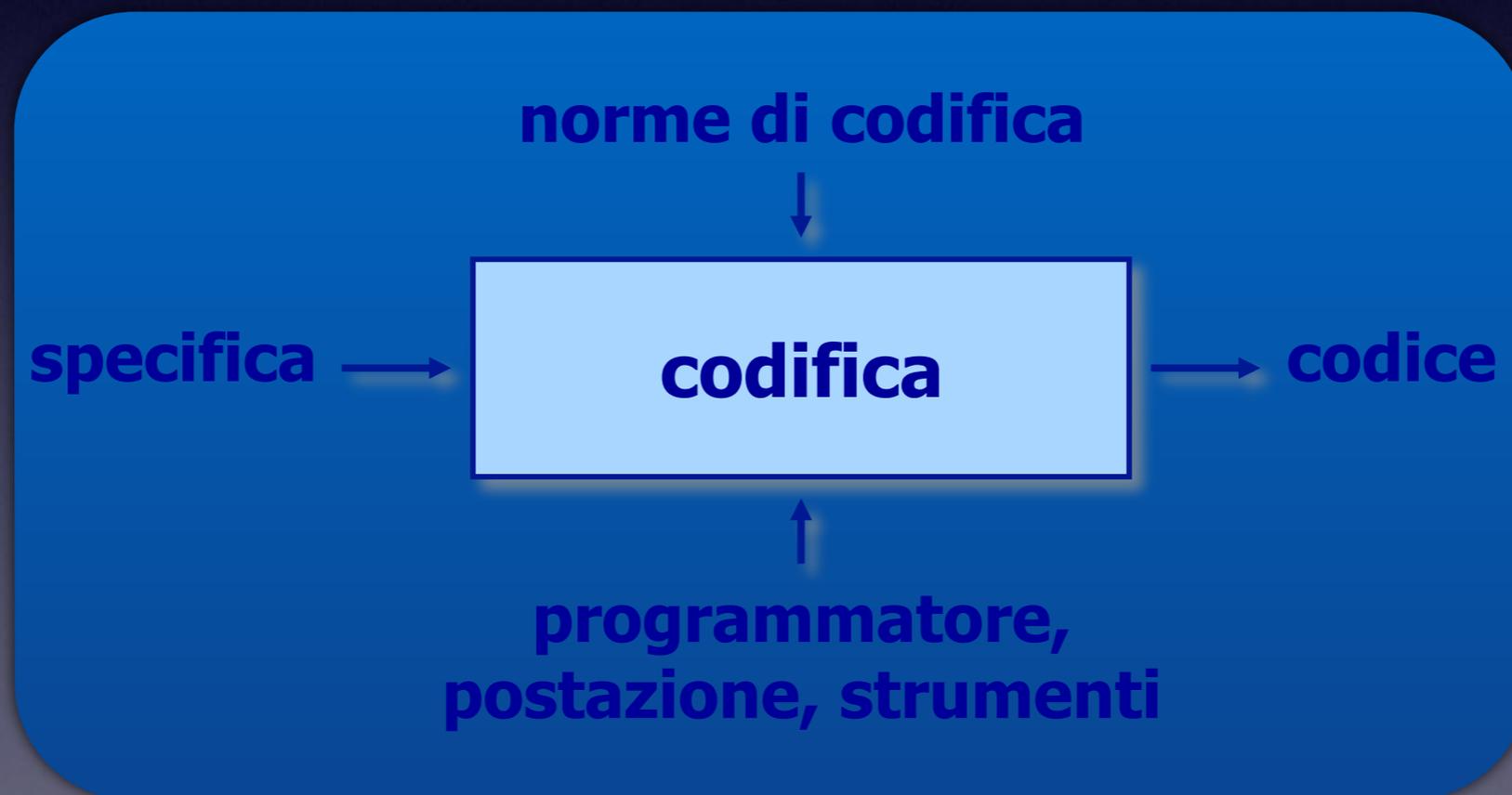
Ingegneria del Software

2. Ciclo di vita

Dipartimento di Informatica
Università di Pisa
A.A. 2014/15

la parola agli standard....

PROCESSO: un insieme di attività correlate che trasformano ingressi in uscite [ISO 9000]



Modellare un processo significa suddividerlo e per ogni attività dire *cosa, quali prodotti, quando*

ISO/IEC 12207:2008

- Processi del ciclo di vita del software
- Il modello più noto e riferito (ne esistono altri)
- Modello ad alto livello
 - Identifica i processi dello sviluppo software
 - Struttura modulare (specializzazione)
 - Identifica le entità responsabili dei processi
 - Identifica i prodotti dei processi

attività primarie

- Comunicazioni
 - con le parti in causa (stakeholder)
 - per capire dominio e prodotto
- Pianificazione
 - costruzione del piano (chi, cosa, quando)
- Modellazione
- Costruzione
 - codice e verifica
- Dispiegamento
 - fornitura al cliente e feedback

di supporto e organizzative

- Controllo del progetto
- Gestione dei rischi
- Valutazione della qualità
- Revisioni tecniche formali
- Misura
- Gestione configurazioni
- Gestione del riutilizzo
- Preparazione e produzione del prodotto
- Gestione dei progetti
- Gestione delle infrastrutture
- Miglioramento del processo
- Formazione del personale

il ciclo di vita

- Modelli del processo primario di produzione sw
 - Concezione, sviluppo, gestione e ritiro
- Identificazione delle attività
 - Modelli generici, indipendenti dal prodotto
- Organizzazione delle attività
 - Ordinamento delle attività
 - Criteri per terminare e passare alla successiva

NON è un
metodo di
sviluppo

NON è un
insieme di
indicazioni
e strumenti

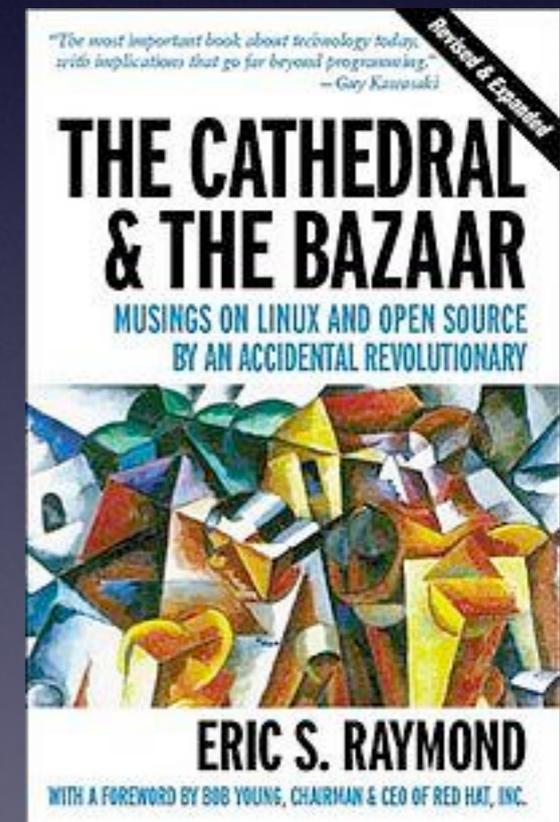
modelli tradizionali

- Code&Fix
 - un non-modello
 - Attività non identificate
 - ...né organizzate
 - Progetti non gestiti
- Modelli organizzati (prescrittivi)
 - Cascata
 - fasi sequenziali, con ritorni e prototipi
 - Incrementali/Evolutivi
 - realizzazione in più passi/ciclici
 - Spirale
 - contesto allargato e modello astratto

modelli di
possibili
processi
standard

modelli “agili”

- Cicli di vita leggeri
 - Flessibilità nell’organizzazione delle attività
 - Meno documentazione (strumentata, nel codice,...)
 - Meno pianificazione (più frequente)
 - Tecniche di gestione del personale
 - Strumenti di sviluppo rapido
- eXtreme Programming, Crystal, Scrum,...



modello a cascata

- Reso “popolare” nel 1970 da Winston Royce

- Successione di fasi sequenziali

- Impossibilità di ritornare a fasi precedenti
- In caso di eventi eccezionali il processo riparte

- Documentazione

- Ogni fase produce “documenti” che la concretizzano
- I documenti sono necessari per la fase successiva
- Modello “document driven”

He was a pioneer in the field of software development, known for his 1970 paper from which the Waterfall model for software development was mistakenly drawn.

[wiki]

caratteristiche delle fasi

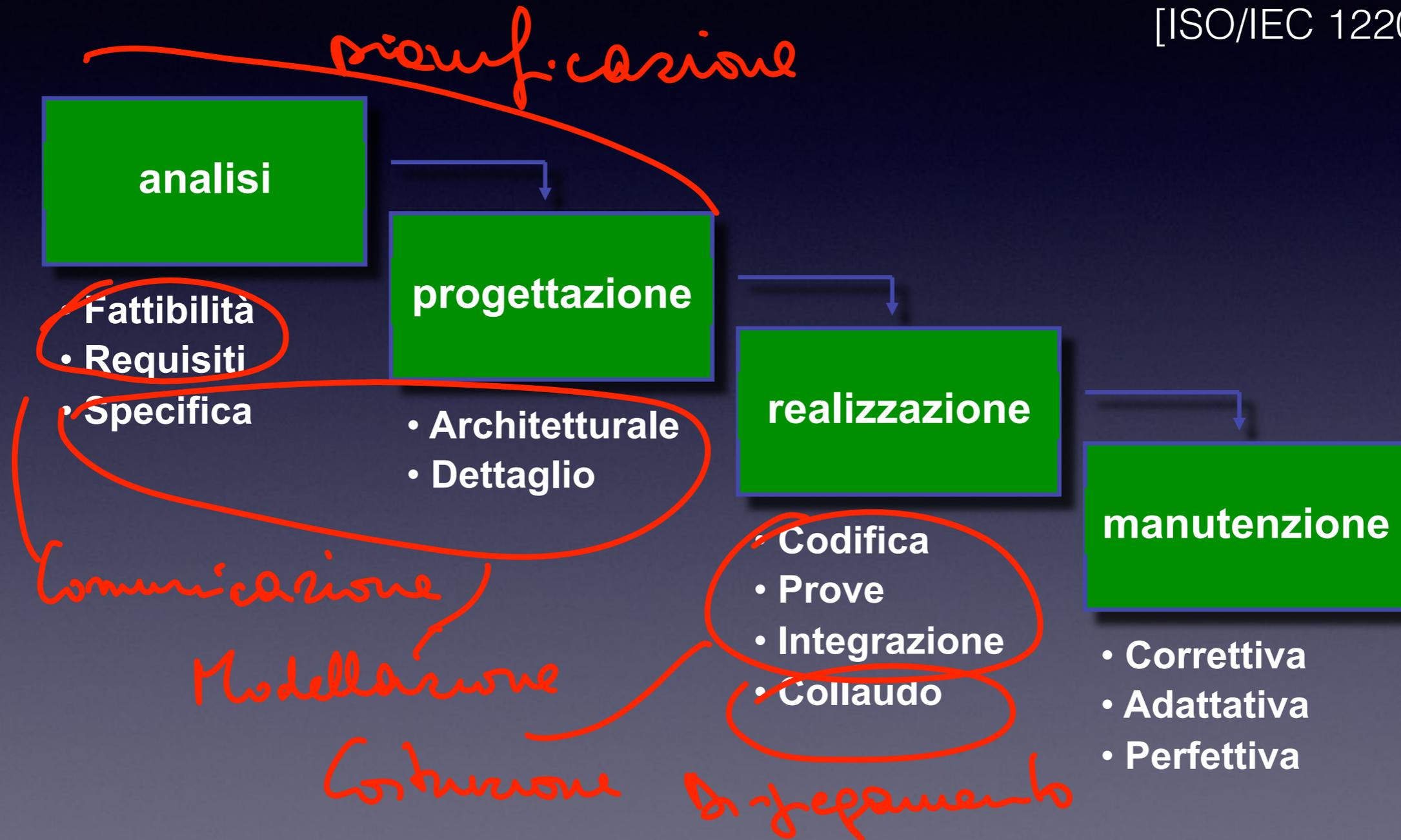
- Le fasi sono descritte in termini di
 - Attività e prodotti intermedi
 - Contenuti e struttura dei documenti
 - Responsabilità e ruoli coinvolti
 - Scadenze di consegna dei documenti
- Dipendenze causali e temporali
- Riferimento per l'identificazione delle attività

schema del modello



in accordo a ISO 12207

[ISO/IEC 12207:2008]



in accordo a Binato *et alii*



variazioni sequenziali

- Mancanza di flessibilità
 - Rigorosa sequenzialità delle fasi
 - Non prevede cambiamenti nei requisiti
 - Genera molta manutenzione
 - Burocratico e poco realistico
- Cascata con prototipazione
- Cascata con ritorni

[i contributi di Royce...]

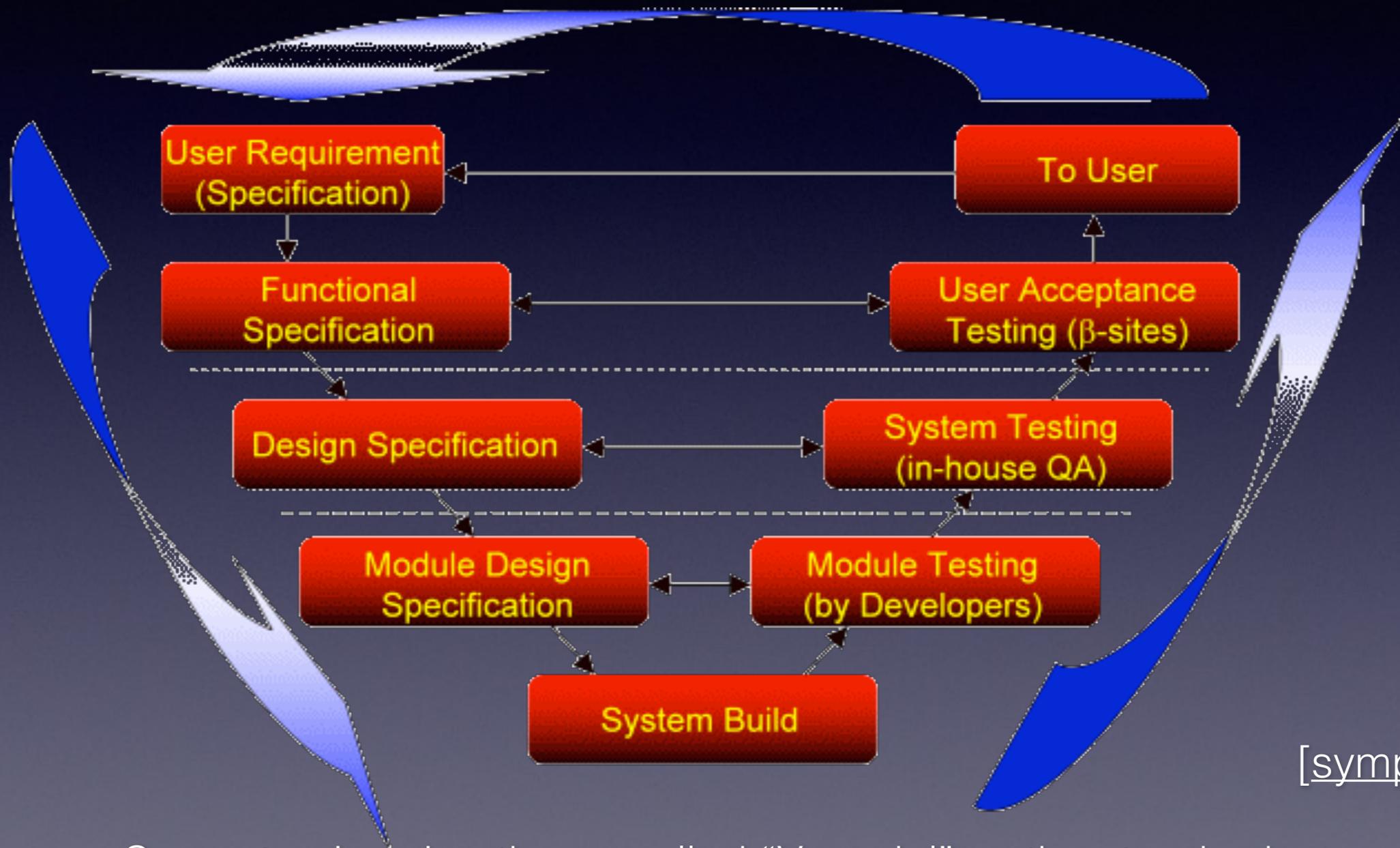
modello a V

The V-Modell is a guideline for the planning and execution of development projects, which takes into account the whole life cycle of the system.

[IABG V1997, V-XT 2005 v-modell.iabg.de]

- Prime formalizzazioni intorno al 1990 [Paul Brook 1986]
- Standard approvato dal governo tedesco
- Enfatica la relazione fra le singole fasi di sviluppo e le corrispondenti fasi di testing

schema “agile”



[sympatec.com]

Sympatec is using the so called “V-model” as the standard life cycle model for all kinds of development projects, i.e. this method is not limited to the development of software

modelli iterativi

- Necessità di modelli adattabili ai cambiamenti
 - delle soluzioni e delle tecnologie
 - dei requisiti
- Soluzione generale
 - Ritardare la realizzazione delle componenti che dipendono criticamente da fattori esterni (tecnologie, hw sperimentale,...)
 - Pianificare le iterazioni

modelli incrementali



- Analisi e progettazione sw
 - Requisiti identificati e stabili
 - Architettura identificata e stabilita
 - I passi della realizzazione incrementale vengono pianificati
- Realizzazione incrementale
 - necessità di “rilasciare” in tempo una parte del prodotto finale
- Sum up: Applicazione iterativa (a release) del modello a cascata

modelli evolutivi



- Analisi preliminare
 - Requisiti *di massima* identificati
 - Architettura *di massima* identificata
 - I passi dell'analisi e realizzazione evolutiva vengono pianificati
- Analisi e realizzazione evolutive
 - raffinamento ed estensione dell'analisi
 - progettazione, codifica, prove e integrazione
- Sum up: versione del modello evolutivo con requisiti instabili

modello a spirale

- Proposto da Barry Boehm nel 1988 [Boehm 2000]
- [A] risk-driven process model generator
 - [T]he risk that technology is unready may be mitigated by an appropriate prototype implementation in an early spiral cycle
 - ...o dominio poco noto, strumenti nuovi, personale non addestrato,...
- *Anchor point milestones* drive the spiral to progress toward completion and offer a means to compare progress between one spiral project and another

schema modello a spirale



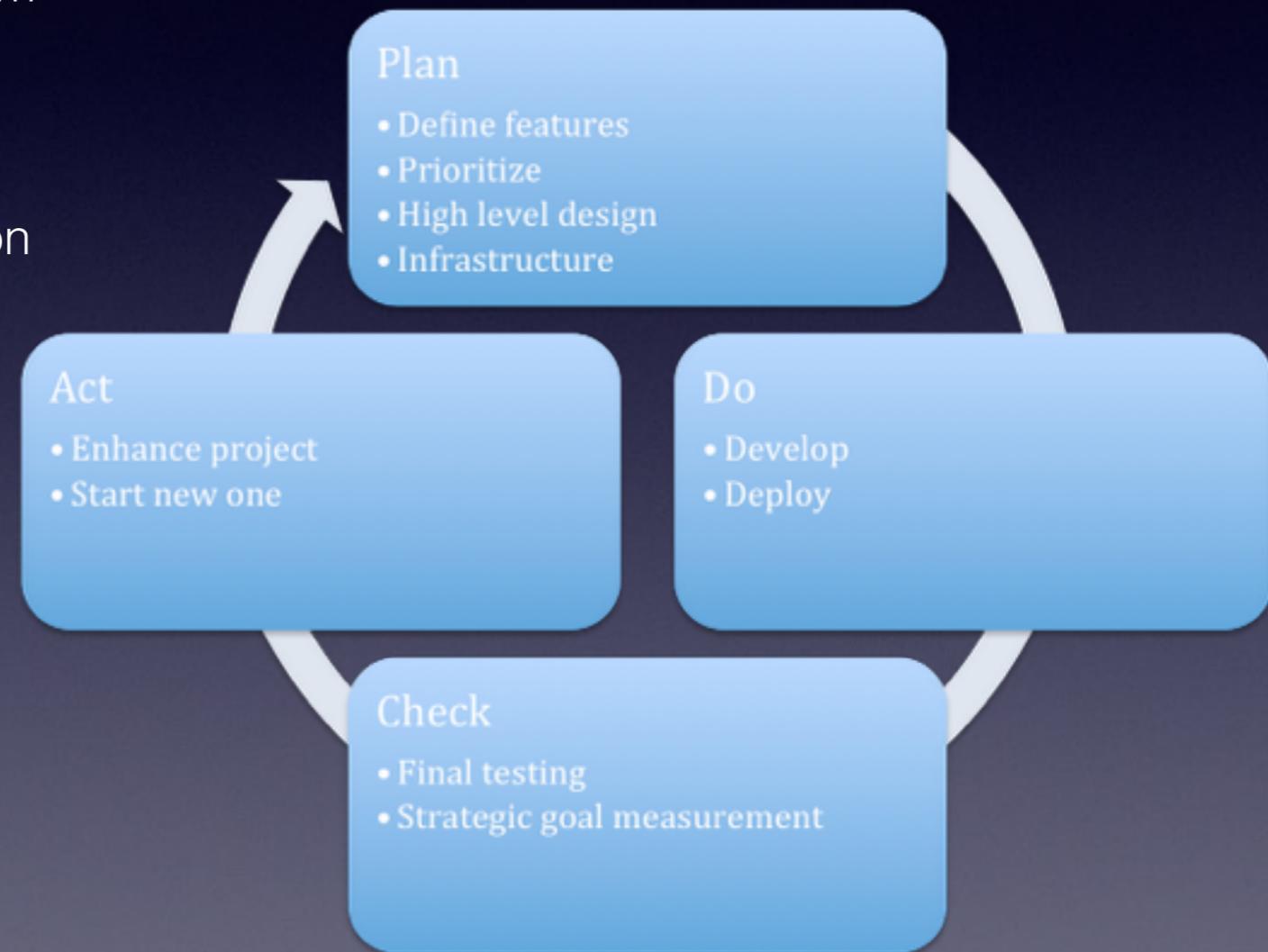
ancora

- Ogni iterazione suddivisa in 4 fasi
- Applicabile ai cicli tradizionali
- Maggior comunicazione e confronto con il committente nelle versioni più recenti

[attenzione ai colori
della Figura 8.3 in
Binato et alii]

ispirato da PDCA/PDPSA

- Plan: Identifying and analyzing the problem
 - sviluppo della policy
- Do: Developing&testing a potential solution
 - deployment della policy
- Check: Measuring how effective the test solution was, and analyzing whether it could be improved in any way
 - Audit, diagnosi, report
- Act: Implementing the improved solution
 - eventuale raffinamento soluzione



[William Edwards Deming 1950]

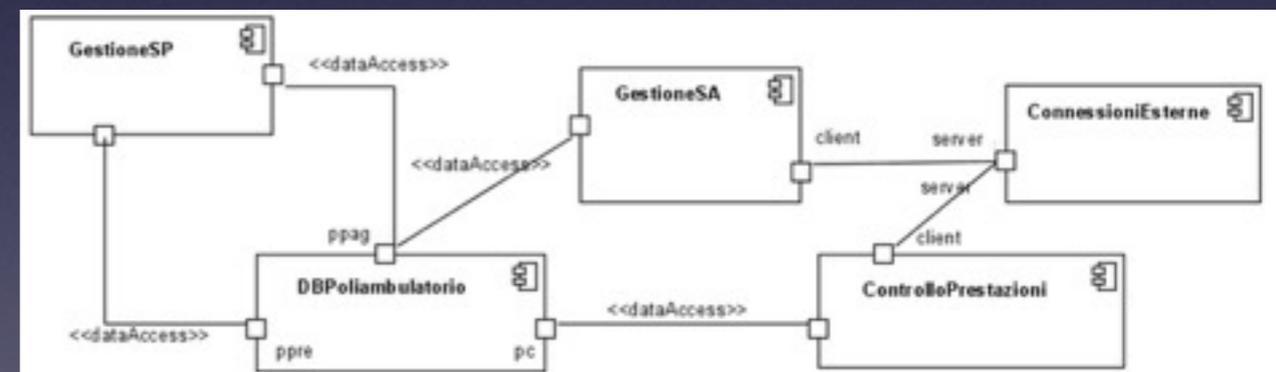
modello a componenti (CBSE)

- Fasi [di un non-modello di successo]

- identificazione dei componenti necessari
- ricerca di componenti comparabili
- eventuale realizzazione di componenti
- integrazione e testing dei componenti

- Favorisce il riuso

- non solo di sw “home-grown”, ma anche terzo (FREE/OS o meno)
- nessuno sviluppa un DBMS, oggi!



[wiki]

[popularised
at Garmish]

unified process (USDP/UP)

- Dai *three amigos* [Booch Jacobson Rumbaugh 1999]
- Guidato dai casi d'uso e dall'analisi dei rischi
 - la raccolta requisiti e i successivi passi sono guidati dallo studio degli use case
- Incentrato sull'architettura
 - UP assegna alla descrizione dell'architettura del sistema un ruolo prioritario. L'approccio si concentra, soprattutto nelle prime fasi, sull'architettura di massima, lasciando i dettagli alle fasi successive. È possibile avere subito una visione generale del sistema facilmente adattabile al cambiamento dei requisiti
- Iterativo incrementale

fasi (temporali) di UP

- Avvio
 - fattibilità; analisi rischi; requisiti essenziali per definire il contesto del sistema; [prototipo]
- Elaborazione
 - analisi requisiti e rischi; sviluppo di un'architettura base; piano per la fase di costruzione
- Costruzione
 - analisi, disegno, implementazione, testing
- Transizione
 - beta testing; aggiustamento delle prestazioni; creazione di documentazione aggiuntiva; attività di formazione; guide utenti; creazione di un kit per la vendita

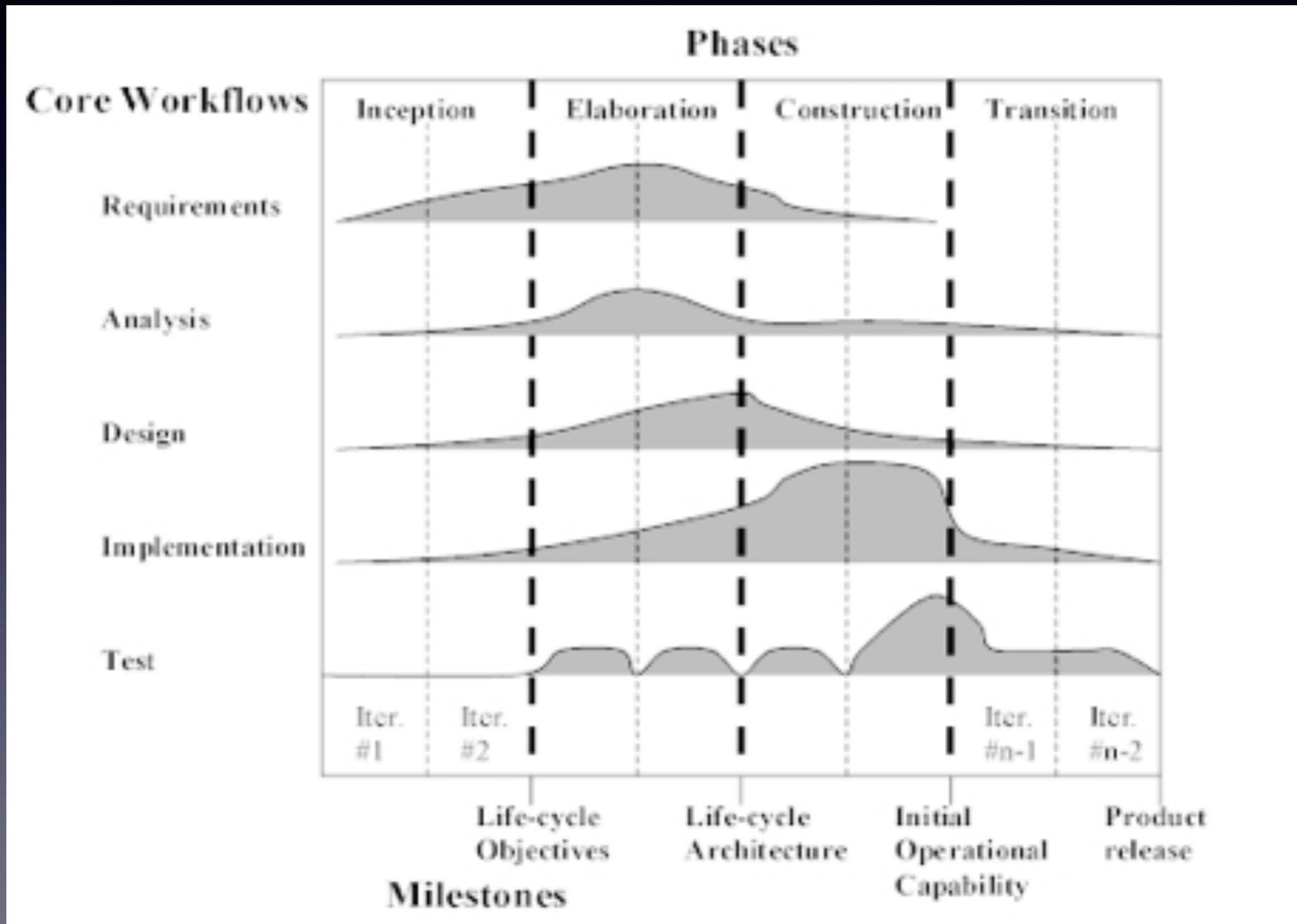
UP rispetto all'incrementale



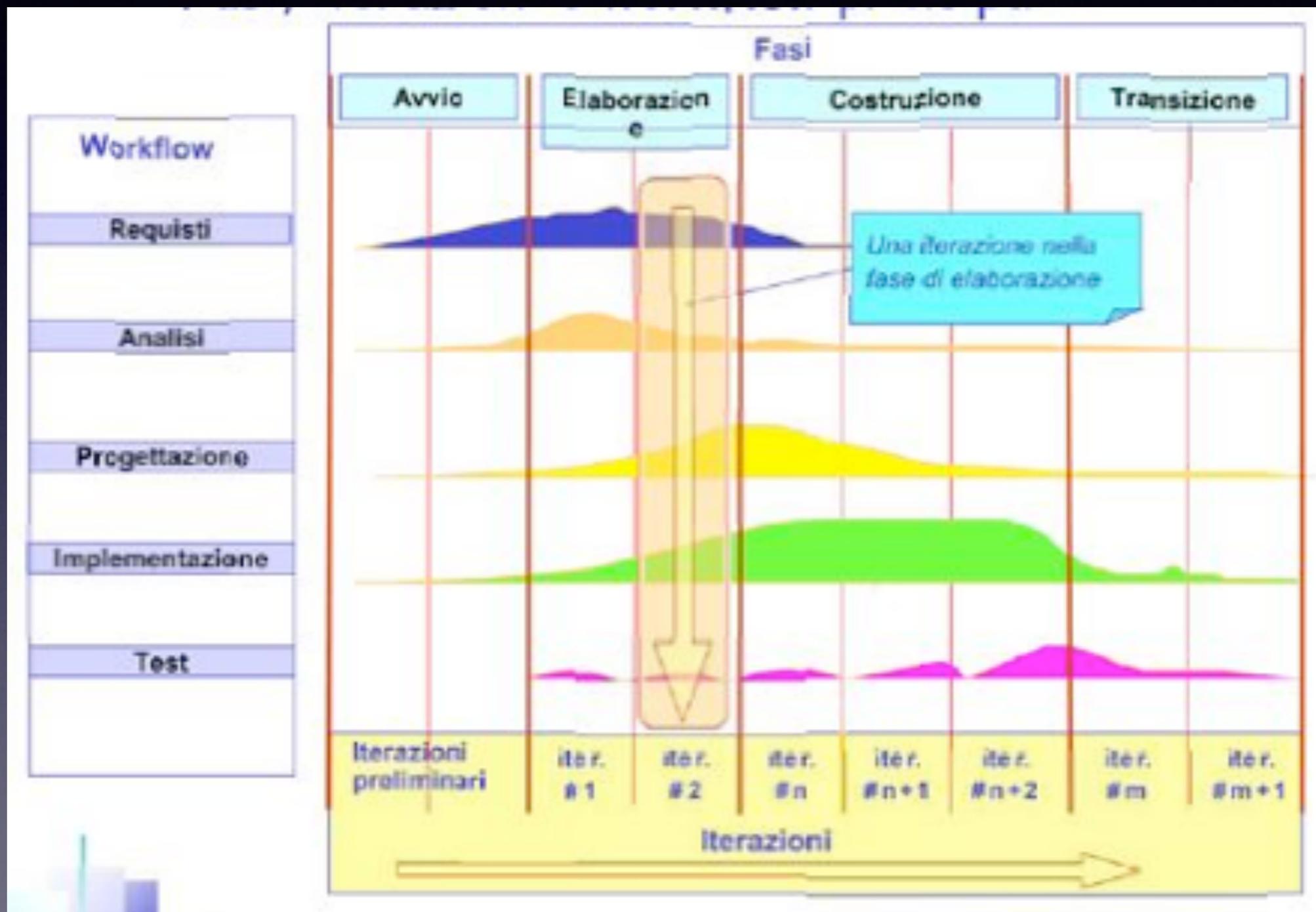
UP in viola, incrementale in giallo

[non necessariamente evolutivo: non è prevista la realizzazione di prototipi!]

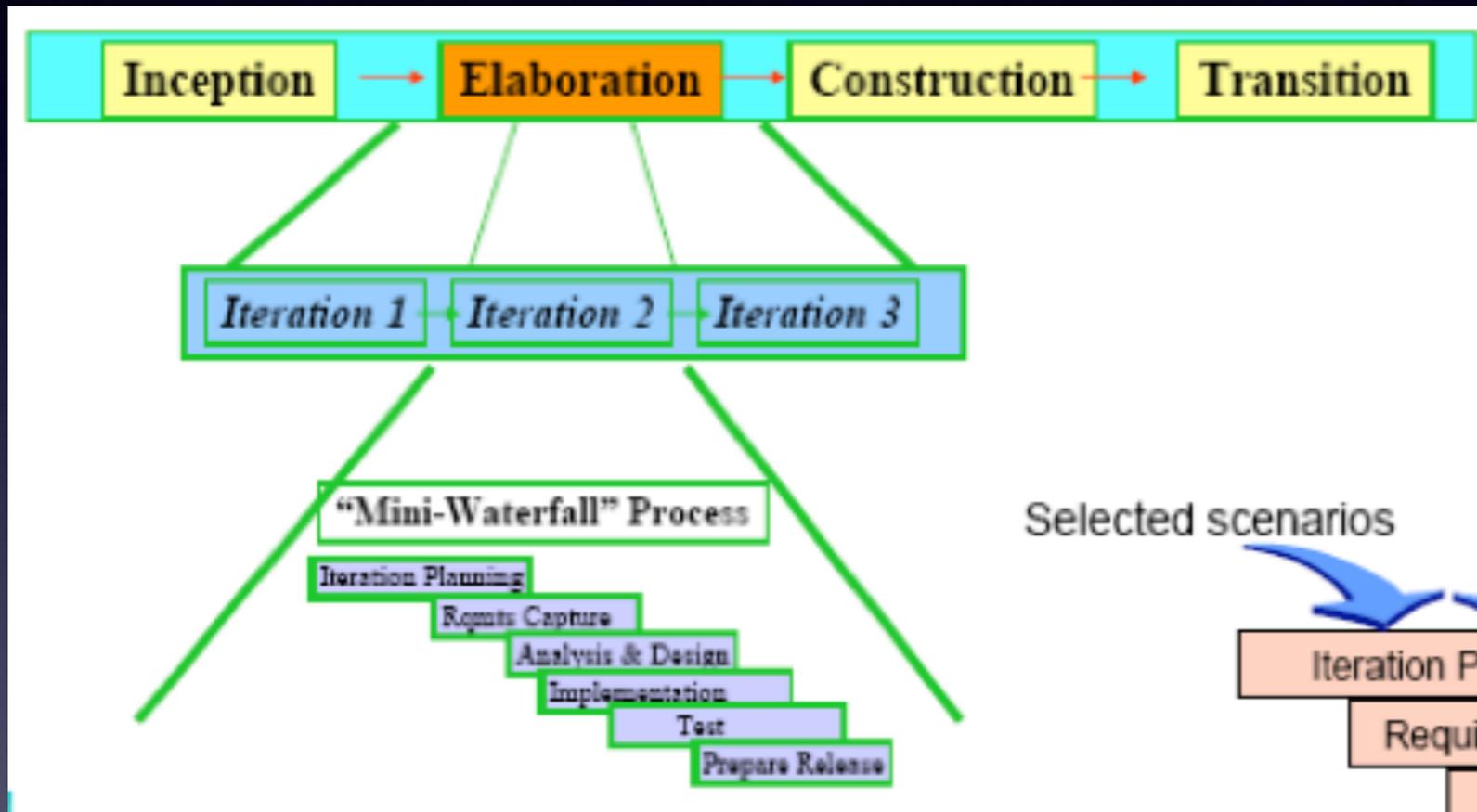
schema di UP



fasi, iterazioni e wf principali



e per ogni fase...



Selected scenarios

- Results of previous iterations
- Up-to-date risk assessment
- Controlled libraries of models, code, and tests



metodi agili

- Particolare metodo per lo sviluppo del software che coinvolge quanto più possibile il committente
- I principi sui quali si basa una metodologia leggera sono solo 4 [manifesto di Snowbird 2001]

[an alternative to documentation driven, heavyweight software development processes]

- Comunicazione

- **le persone e le interazioni sono più importanti dei processi e degli strumenti**
[tutti possono parlare con tutti, persino l'ultimo dei programmatori con il cliente]
[le relazioni e la comunicazione tra gli attori di un progetto sono la miglior risorsa]
- **bisogna collaborare con i clienti al di là del contratto**
[la collaborazione diretta offre risultati migliori dei rapporti contrattuali]

metodi agili

- Semplicità [la descrizione formale sia il più semplice e chiara possibile]
 - **è più importante avere software funzionante che documentazione**
 - mantenere il codice semplice e avanzato tecnicamente, riducendo la documentazione al minimo
 - Feedback
 - bisogna rilasciare nuove versioni del software ad intervalli frequenti
 - sin dal primo giorno si testa il codice
 - Coraggio
 - si dà in uso il sistema il prima possibile e si implementano i cambiamenti richiesti man mano
 - **bisogna essere pronti a rispondere ai cambiamenti più che aderire a un progetto**
 - Adatti a progetti con meno di 50 persone
- agilemanifesto.org/

un metodo agile: XP

- **Una prassi implementativa**
- Pianificazione flessibile
 - basata su scenari proposti dagli utenti
 - ...e che coinvolge i programmatori
- Rilasci frequenti
 - due-quattro settimane
 - inizio di una nuova pianificazione
- Progetti semplici
 - comprensibili a tutti
- Metafore condivise
 - per descrivere una funzionalità del sistema si usa una metafora. [Aiuta a far sì che il team usi un sistema comune di nomi di entità, tale che sia immediato trovare, per uno sviluppatore, un modulo in base al nome, o sia chiaro dove inserire le nuove funzionalità appena sviluppate.]
- Verifica (testing)
 - di unità e di sistema (basata su scenari)
 - supporto automatico
 - Test Driven Development [casi di test = specifica]
 - cliente sempre a disposizione (circa ogni settimana)

un metodo agile: XP

- Programmazione a coppie
 - un solo terminale: il *driver* scrive il codice mentre il *navigatore* controlla il lavoro del suo compagno in maniera attiva
- Code Refactoring
 - modifiche senza cambi nel comportamento
 - “se un metodo necessita di una spiegazione, riscrivilo!” [codice *auto-esplicativo*]
- Collettivizzazione del codice
 - accesso libero
 - integrazione continua
 - standard di codifica
- No lavoro straordinario
- Daily Stand Up Meeting

I think that ultimately, Extreme Programming has mushroomed in use and interest, not because of pair-programming or refactoring, but because, taken as a whole, the practices define a developer community freed from the baggage of Dilbertesque corporations

una rassegna

