

presentare o dirigitli presentando a info essendi

MODELLO : strategia rappresentativa per rispondere a domande (WHY, WHAT, HOW)

MODELLO DI CICLO DI VITA : (in direzione, non è statico)

esempio

- ANALISI DEI REQUISITI
- ↓
- DESIGN
- ↓
- IMPLEMENTATION
- ↓
- TESTING
- ↓
- EVOLUTION

ITERAZIONE

⋮

MODELLO SEQUENZIALE : catena di montaggio

1° modello è : SEQUENZIALE (A CASCATA)

(nuovo passo) PRE CONDIZIONE : dove essere vero prima dell'azione (ponte aperto)
POST CONDIZIONE : dove essere vero prima dopo l'azione (scatto partenza quelle cose sono vere)

↓

si scansa software il più tardi possibile (dopo le PRECONDIZIONI)

ANALISI DEI REQUISITI

(finisce quando l'analisi è documentata)

ante soluzione dove (si passa a)

DESIGN

documento con diagrammi

BIG-BRANCH INTERSECTION

però il software scritto lo mettiamo insieme in una volta sola

IMPLEMENTATION

SOLO IN AUMENTI

→

CONVENZIONE
MODIFICATIVA
EVOLUTIVA

MANUTENZIONE

↓
I REQUISITI NON POSSONO PIÙ CAMBIARE! È LA FINE DEL PROJECT
MA IL SOFTWARE È MANUTENIBILE (SOFT)

TRAPPOLO
ALGIDIO

↓
I REQUISITI POSSONO NON ESSERE PRECISI
E CAMBIARE

MODELLO INTUITIVO (intuitive) \rightarrow pittore
 (while) \downarrow
 OLFUSO!
 ni posso arrivare o allentare il risultato
 (ma so quando ci riuscirò), opportuno
 un provvedimento
 cambio di ogni passo ma mi posso
 allentare

MODELLO TECNICO
 (to do, debito con interessi)
 è inevitabile ma è da seguire
 è difficile ma è a dare tentare

MODELLO INCREMENTALE: \rightarrow scultore
 (osservazioni al far per iterare) \rightarrow possiamo sapere di più il risultato
so quanti passi fare
 sapendo quanti passi fare

\downarrow
 si lavora su requisiti + important
 (il resto nel to do)
 \downarrow
 spazio di design che si aspetta
 aggiunte

\rightarrow modo per avvicinarsi al modello incrementale (modiso)
MODELLO AGILE:
 (agile, non rigido)
 \rightarrow SCRUM:
 • backlog (to do organizzato, etichettato)
 (si svuota e si riempie)
 • sprint (ciclo ripetitivo, durata standard)

Sviluppo \neq sempre complete

Ciclo di vita \neq sviluppo

\downarrow
 continuo che
 non si chiude mai
 (software vive in un
 certo stato)

MANO DI AGIRE

- **SYSTEMATIC** : ripetibile
- **DISCIPLINATO** : seguire regole (way of working)
- **QUANTIFICABILE** : dare priorità di quanto ci vorrà
(quanti requisiti soddisfatti?)

PROTOTIPO : realizzare qualcosa che assenti a dare
della natura e mi dice se è fattibile
(ce la posso fare?, come?)

→ **USA & CRITA** : lo faccio per imparare cose (mi è
(POC) il prodotto), senza contenere rischi
(mi dico se la direzione è ragionevole)

→ **INCREMENTAL** : prototipo su cui aggiungo pezzi,
(product baseline) aggiungo il minimo punto di fattibilità
(punto di partenza)
(MVP, minimum viable product)

PER EVANGELICARE USIAMO

→ **SCORE CARD** : a che punto siamo con:

REQUIREMENTS (6 assi)

- concerned (visione comune con il proprietario)
- bounded (limiti)
- coherent (coerenti tra loro)
- Acceptable (sono d'accordo con i requisiti)
- Addressed (soddisfatti)
-

WAY OF WORKING

- Principles stabiliti (abbiamo deciso le regole di come comportarci)
- Functions stabiliti (abbiamo deciso i blocchi, documentato il way of working)

- In use (lo prendiamo, il treno sui binari)
(come se fanno, da migliore ...)

• 2^a place

• working well

• retired