

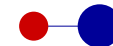
Stato dell'Ingegneria del Software: Problemi e Pratiche

I Tipi di Prodotti Software

Il Software...

PROGRAMMA

- ⇒ È un **insieme di istruzioni** autoconsistenti rispetto ad uno o più obiettivi.
- ⇒ Spesso è **usato dallo stesso** autore. E' difficile farlo utilizzare da altri perché per **scarsa documentazione** è difficilmente comprensibile
- ⇒ I suoi **difetti** sono **rilevati**, in genere, **sul campo** perché è **scarsa** la **validazione** durante la sua produzione.
- ⇒ Ha **vita** relativamente **breve** perché la manutenzione fa decadere la sua qualità e diventa sempre più **difficile e costoso farlo evolvere**.



... Il Software ...

APPLICAZIONE

- ⇒ È un **insieme di programmi interagenti** tra loro.
- ⇒ Spesso è venduto come un **pacchetto usabile** da persone che non hanno dimestichezza con l'informatica perché sono **forniti, almeno, di un'interfaccia e di documentazione d'uso**.
- ⇒ I loro **difetti** sono scoperti essenzialmente dagli utilizzatori, ma durante la produzione una parte **sono scoperti attraverso la validazione**.
- ⇒ Spesso hanno **bassi livelli di qualità**; sono poco attrezzati per il trasferimento a nuovi sviluppatori; pertanto la loro qualità decade rapidamente e diventa sempre più **costosa e rischiosa la loro manutenzione**.

... Il Software

SISTEMA SOFTWARE

⇒ È un insieme di programmi interagenti che copre un Dominio Applicativo con

- alto livello di qualità,
- completo di tutta la **documentazione** : i **requisiti**, la **progettazione** che spiega la sua struttura e le decisioni che hanno giustificato la sua strutturazione; i **manuali d'uso**.

⇒ Per far **rientrare gli alti costi di produzione**, essi sono destinati ad un esteso bacino di utenza anche con piattaforme diverse, pertanto: la **usabilità** e la **portabilità** sono caratteristiche chiave.

⇒ Per poter **essere redditizio**, un sistema software deve **avere una lunga vita** e deve **invecchiare lentamente**.

Applicazioni per l'Impresa ...

⇒ Applicazioni per l'impresa sono Sistemi Software caratterizzati da :

- **Dati persistenti**, necessari per passare le informazioni tra differenti applicazioni e tra differenti esecuzioni della stessa applicazione
- **Rilevanti volumi di dati**, un sistema di medie dimensioni potrebbe avere diversi GB di dati organizzati in decine di tipi di records ed in milioni di records

... Applicazioni per l'Impresa

- **Accesso concorrente ai dati**, da diversi utilizzatori dell'applicazione
- **Rilevante numero di schermate componenti l'interfaccia**, è frequente avere interfacce con centinaia di schermate
- **Integrazione tra applicazioni che supportano domini differenti della stessa impresa**, molto spesso costruite con approcci eterogenei

Problemi della Ingegneria del
Software

Dissonanze Concettuali

- ⇒ Molti concetti del Dominio Applicativo sono interpretati in modo diverso da utenti e da applicazioni diversi.
 - Per esempio, un **cliente** può essere considerato:
 - un soggetto con cui l'impresa ha una **relazione commerciale corrente**, oppure
 - un soggetto con cui si è **intrattenuta una relazione commerciale**, anche se tale relazione non è stata mantenuta

Struttura Complessa

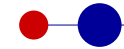
- ⇒ Un **Dominio Applicativo** include, in genere, un insieme di **Processi di Business che hanno molte relazioni tra loro**
- ⇒ Le **Applicazioni d'Impresa** devono tener conto delle molte **interrelazioni tra i processi di business** e perciò risultano essere molto complesse.

Prestazioni...

- ⇒ **Tempo di risposta**, tempo necessario per elaborare un processo: da quando si pressa il tasto di avvio a quando mostra il risultato
- ⇒ **Reattività**, tempo che impiega il sistema a notificare che ha acquisito la richiesta dell'utente, e' indipendente dal tempo di risposta

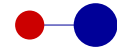
...Prestazioni...

- ⇒ **Latenza**, il tempo minimo necessario per ricevere un qualunque tipo di risposta, compresa la notifica che l'applicazione non è riuscita ad eseguire nulla;
- ⇒ **Troughput**, quanto lavoro il sistema riesce ad eseguire nell'unità di tempo: transazioni per secondo; byte trasferiti per secondo...



... Prestazioni...

- ⇒ **Carico**, quanto lavoro contemporaneo può fare il sistema: il numero di utenti che possono lavorare contemporaneamente
- ⇒ **Sensibilità al carico**, quanto varia una prestazione con il cambiamento di carico
- ⇒ **Efficienza**, la prestazione rapportata alle risorse utilizzate: throughput/numero di CPU

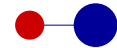


... Prestazioni

- ⇒ **Capacità**, il massimo throughput o carico che un sistema può reggere. Può essere un massimo assoluto od una soglia oltre la quale le prestazioni del sistema possono decadere sensibilmente
- ⇒ **Scalabilità**, misura della capacità di modificare le prestazioni del sistema con l'aumento delle risorse disponibili



Pratiche



Vecchie Pratiche...

- ⇒ Strutturazione del software per facilitare la manutenzione
 - Principi di Parnas circa **l'incapsulamento dell'informazione/information hiding**
 - Diagonalizzazione della matrice **Requisiti x Componenti**



... Vecchie Pratiche: Matrice Requisiti x Componenti...

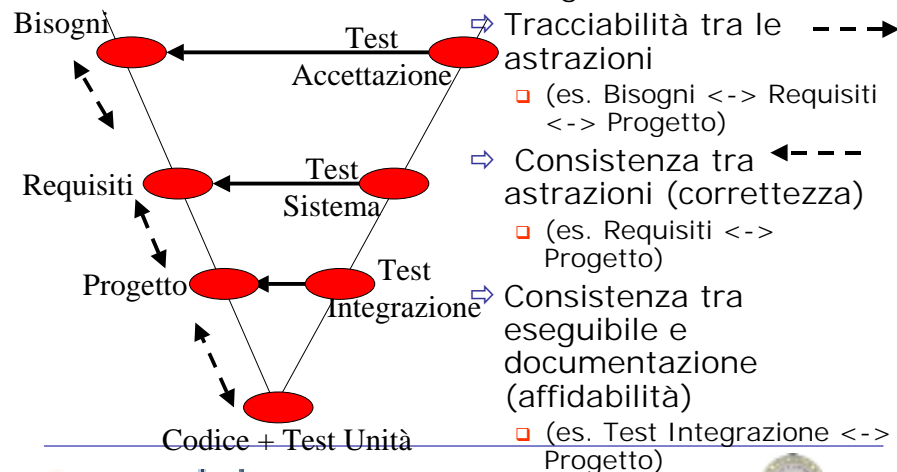
	R ₁	R ₂	R ₃	...	R _{n-1}	R _n
C ₁	X	X	X		X	X
C ₂	X		X		X	X
C ₃		X	X			X
⋮						
⋮						
⋮						
C _{m-1}	X				X	X
C _m	X	X			X	X

- ⇒ Una matrice diagonale descrive un'applicazione che
 - Facilita la validazione
 - facilita la manutenzione
 - facilita la stima dei costi di manutenzione
- ⇒ Le **ispezioni** sul progetto aiutano a migliorare questa matrice

... Vecchie Pratiche...

- ⇒ La **documentazione** come manufatto focale dello sviluppo di un'applicazione
 - La documentazione deve essere **tracciabile verticalmente ed orizzontalmente**
 - La tracciabilità è **assicurata** se il codice non è documentato a posteriori ma è ricavato dalla documentazione

... Vecchie Pratiche : Modello di prodotto tracciabile



Problemi aperti della Ingegneria del Software

- ⇒ La **carenza di conformità** dei processi software con queste pratiche causa lo sviluppo di **applicazioni difficili da mantenere**
- ⇒ E' necessaria la raccolta ed il trasferimento agli sviluppatori di **evidenze sperimentali** circa:
 - la relazione tra questi principi e la economicità di costruzione, distribuzione e manutenzione del software

Nuove Frontiere della Ingegneria del Software

- ⇒ Potenziamento dello sviluppo attraverso la distribuzione della produzione
- ⇒ Sviluppo di Software Open Source
- ⇒ eXtreme Programming
- ⇒ Software ad alta sicurezza ed affidabilità
- ⇒ Linee di Prodotto
- ⇒

Potenziamento dello sviluppo attraverso la distribuzione della produzione...

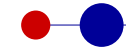
- ⇒ Obiettivi
 - far concentrare l'impresa su poche competenze centrali
 - utilizzare le risorse umane disponibili sull'intero globo

... Potenziamento dello sviluppo attraverso la distribuzione della produzione ...

- ⇒ Problemi
 - Specifiche del prodotto devono essere chiare, altrimenti si costruisce un'applicazione difforme dai requisiti
 - Incompatibilità della maturità dei processi delle imprese che cooperano fa rischiare la **violazione dei tempi e dei costi previsti** per la produzione e la manutenzione.

... Potenziamento dello sviluppo attraverso la distribuzione della produzione

- ⇒ Suggerimenti
 - Solo le imprese che sanno specificare possono utilizzare la sub contrattazione
 - Se un'impresa non ha profonda competenza nella gestione dei progetti co-locali non può pensare a distribuirli fuori casa



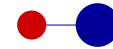
Sviluppo di Software Open Source...

⇒ Obiettivi

- Utilizzare la creatività di comunità estese
- Utilizzare la potenza di sviluppo di una grande comunità

⇒ Problemi

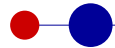
- Non è chiaro e non è documentato lo **scopo della documentazione**
- Non è chiara la **responsabilità della correzione e della evoluzione del software**



... Sviluppo di Software Open Source

⇒ Suggerimenti

- E' necessario accordarsi con gli sviluppatori perché applichino le buone pratiche dell'ingegneria del software



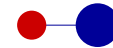
eXtreme Programming...

⇒ Obiettivi

- Riduzione dei tempi e dei costi di sviluppo
- Utilizzazione di una buona pratica dell'Ingegneria del Software: **la ispezione**
- Evitare l'uso di processi **prescrittivi e pesanti**

⇒ Problemi

- Gli sviluppatori sembrano molto spesso simili agli **hacker**; questo processo si sta diffondendo in ambienti poco maturi
- il processo non consente di patrimonializzare i risultati per riusarli in progetti diversi



... eXtreme Programming

⇒ Suggerimenti

- si deve adottare con chiarezza di scopo, dipendentemente dal dominio di applicazione e dagli obiettivi del progetto;
- deve essere utilizzato con modalità **orientata al prodotto** e non al processo
- non deve indurre ad abbandonare l'orientamento al processo ma deve essere utilizzato come un **processo leggero**

Linee di prodotto ...

⇒ Obiettivi

- Utilizzare le somiglianze tra sistemi software differenti per migliorare lo sviluppo e la manutenzione delle applicazioni
 - tempi e costi di sviluppo
 - qualità

... Linee di prodotto ...

⇒ Problemi

- La costituzione di una famiglia di applicazioni che rappresentino una linea di prodotto richiede maggiore tempo della costruzione di una singola applicazione; perciò il **processo di costruzione della linea di prodotto è incrementale**;
- Il primo membro di una linea utilizza una piattaforma ed un modello di dominio applicativo di riferimento; la **linea si popolerà con la necessità di cambiare la piattaforma o di evoluzione del modello di dominio o dello stesso dominio**;

... Linee di prodotto ...

⇒ Suggerimenti

- Le linee di prodotto sono un approccio che promuove attivamente il riuso
- la efficacia migliora incrementalmente, con l'apprendimento di nuove diversità dei prodotti nello stesso dominio;

... Linee di prodotto

⇒ Esperienze

- Le esperienze del SEI e dello IESE suggeriscono che l'investimento è giustificato dall'economia di produzione e manutenzione
- Il ritorno dell'investimento si ha con il terzo sistema della stessa linea di prodotti

Software ad Alta Sicurezza ed Affidabilità...

⇒ Obiettivi

- Certificazione dei prodotti software con continuità, prima e dopo ogni intervento di manutenzione

⇒ Problemi

- Enorme **sforzo e costo di certificazione**, soprattutto per sistemi la cui manutenzione è affidata a terzi
- La certificazione per prodotti mantenuti localmente richiede la **incrementalità delle ispezioni & test**

... Software ad Alta Sicurezza ed Affidabilità

⇒ Suggerimenti

- Le applicazioni devono essere costruite per essere modificate facilmente, utilizzando tutti i principi di strutturazione della ingegneria del software
- Migliorare l'uso di tecniche di manutenzione, verifica e validazione incrementali

⇒ Esperienze

- La linea di prodotti è un approccio adeguato per il superamento di questi problemi perché è massimizzato il riuso e, quindi, è più facile la certificazione incrementale.