Processo – parte IV

Leggere Cap. 21 Sommerville, Sez. 7.5 Ghezzi et al.

università di ferrara
DA SEICENTO ANNI GUARDIAMO AVANTI.

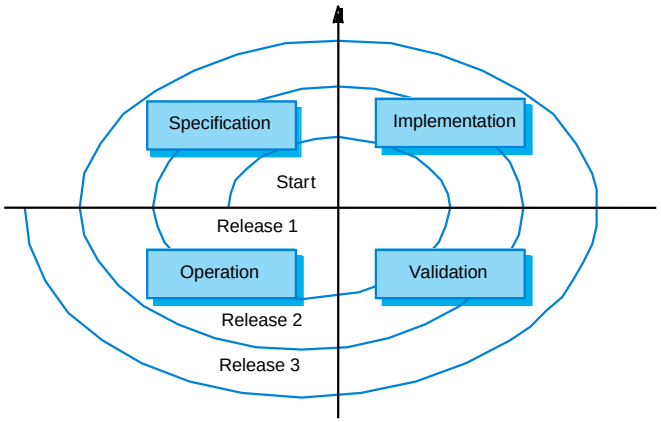
Cambiamenti del software

- Sono inevitabili:
  - gli errori devono essere corretti
  - emergono nuovi requisiti
    - funzionali
    - non-funzionali
  - l'ambiente di lavoro cambia
  - cambiano le piattaforme hardware e software
- La gestione dei cambiamenti del software è un'attività critica di molte organizzazioni

Evoluzione del software

- Le organizzazioni fanno investimenti ingenti nei loro sistemi software
- Per mantenere i sistemi software funzionali e produttivi, è necessaria una manutenzione continua
- Generalmente, le organizzazioni investono più nella manutenzione che nello sviluppo di nuovi software.

Modello a spirale dell'evoluzione del software



Dinamiche di evoluzione di un sistema software

- Studio dei processi di modifica di un sistema software
- In anni di studi, Lehman e Belady hanno individuato “leggi” che governano l’evoluzione.
- Applicabili soprattutto a sistemi di grandi dimensioni, sviluppati da/per grandi organizzazioni

Leggi di Lehman

- Modifiche continue: un programma utilizzato in un ambiente reale deve necessariamente cambiare nel tempo, o diventerà progressivamente meno utile.
- Complessità crescente: quando un programma in evoluzione cambia, la sua struttura tende a diventare più complessa. Si devono dedicare risorse aggiuntive per preservare e semplificare la struttura.
- Evoluzione dei grandi programmi: L’evoluzione dei programmi è un processo autoregolato: gli attributi di sistema come la dimensione, il tempo tra una release e l’altra e il numero di errori rilevati sono approssimativamente invarianti per ciascura release di sistema.

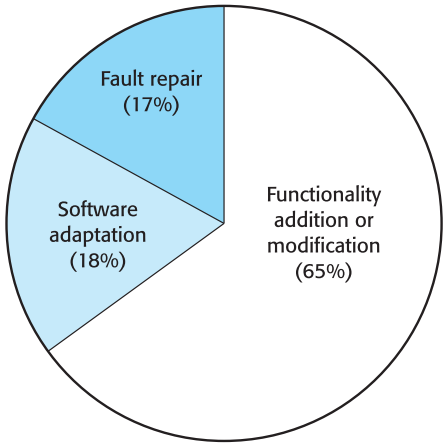
Leggi di Lehman

- Stabilità organizzativa: durante il ciclo di vita di un programma il suo tasso di sviluppo è approssimativamente costante e indipendente dalle risorse dedicate allo sviluppo stesso.
- Conservazione della familiarità: durante il ciclo di vita di un programma le modifiche incrementali per ogni release sono approssimativamente costanti.
- Crescita continua: le funzionalità offerte dai sistemi devono crescere continuamente per mantenere soddisfatti gli utenti.
- Qualità deteriorata: la qualità dei sistemi apparirà deteriorata se non vengono adattati ai cambiamenti del loro ambiente operativo.

Manutenzione

- Tre tipi di manutenzione:
  - correttiva: corregge gli errori presenti
  - adattativa: modifica il software per permetterne il funzionamento quando cambiano le condizioni operative
  - perfettiva: modifica il software per migliorarne alcune qualità o rispondere a nuovi requisiti.
- Inevitabile per mantenere la funzionalità di un sistema software

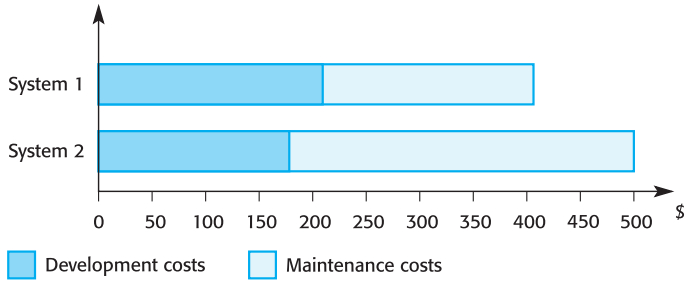
Distribuzione dei costi di manutenzione



Costi di manutenzione

- Generalmente più alti dei costi di sviluppo.
- Determinati da fattori tecnici e non.
- Aumentano col tempo: la manutenzione corrompe il software, rendendo la ulteriore manutenzione più difficile.
- Il software datato può avere costi di manutenzione più alti (linguaggi obsoleti, strumenti di sviluppo di nicchia)

Costi di sviluppo e di manutenzione



- Investire durante lo sviluppo per un prodotto più facilmente evolvibile può ridurre drasticamente i costi di manutenzione

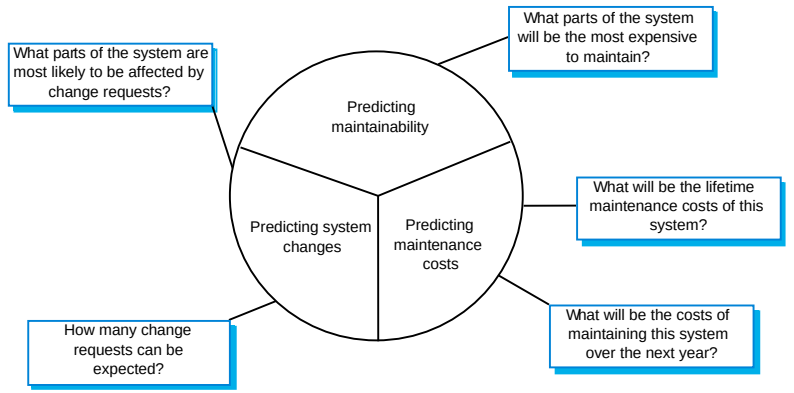
Motivi del maggior costo della manutenzione

- Stabilità del team
  - Spesso il team di sviluppo viene sciolto alla fine dello sviluppo stesso
- Responsabilità contrattuale
  - Chi sviluppa un sistema può non avere responsabilità nella manutenzione, quindi non è incentivato a investire nella manutenibilità
- Capacità dello staff
  - Chi si occupa di manutenzione è spesso meno esperto di chi si occupa di sviluppo.
- Età e struttura del programma
  - Col tempo e con le successive modifiche, la struttura del programma si corrompe e diventa più difficile da capire e da modificare.

Previsione della manutenzione

- Consiste nell'individuare le parti del sistema che possono creare problemi e avere alti costi di manutenzione.
  - L'accettazione di una modifica dipende dalla manutenibilità dei componenti interessati
  - Le modifiche al sistema ne riducono la manutenibilità
  - I costi di manutenzione dipendono dal numero di modifiche e il costo di una modifica dipende dalla manutenibilità dei componenti interessati.

Previsione della manutenzione



Numero di richieste di modifica

- Dipende dalle relazioni fra il sistema e il suo ambiente:
  1. Numero e complessità delle interfacce del sistema
  2. Numero dei requisiti di sistema intrinsecamente volatili
  3. Processi aziendali in cui il sistema viene utilizzato

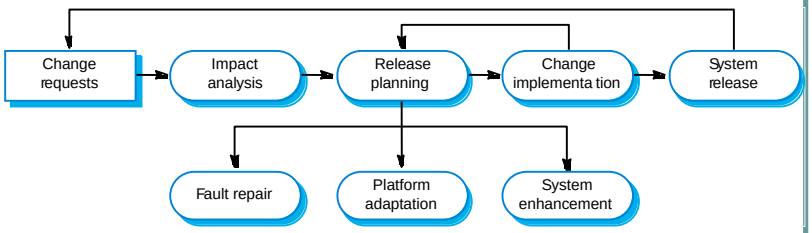
Manutenibilità del sistema

- Dipende dalla complessità dei componenti del sistema
- Gli studi dimostrano che i costi di manutenzione si concentrano su un numero relativamente piccolo di componenti
- La complessità dipende da
  - Complessità delle strutture di controllo
  - Complessità delle strutture di dati
  - Dimensione di oggetti, procedure e moduli.

Misure per la valutazione della manutenibilità

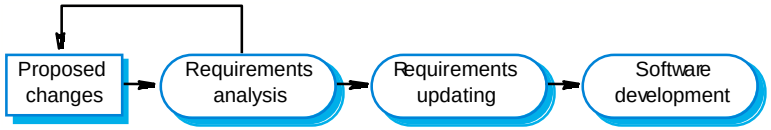
- Utilizzabili quando il sistema è già in funzione:
  - Numero di richieste di manutenzione correttiva
  - Tempo medio richiesto per l'analisi dell'impatto
  - Tempo medio richiesto per implementare una modifica
  - Numero di richieste di modifica in attesa
- Un aumento di una di queste grandezze indica un calo della manutenibilità

Processo di evoluzione del sistema



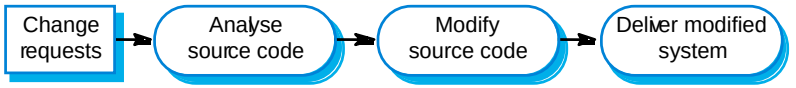
- Iterazione del processo di sviluppo
- Modelli evolutivi

Implementazione delle modifiche



- Le modifiche
  - partono dai requisiti
  - si propagano alla progettazione
  - si propagano all'implementazione
- Test di regressione

Modifiche urgenti

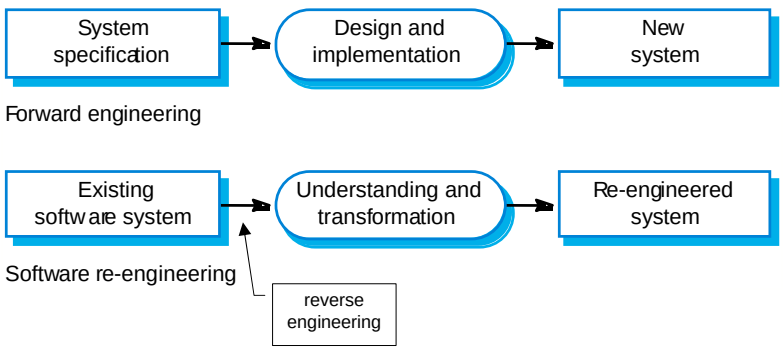


- Intervento rapido, necessario per
  - Gravi malfunzionamenti
  - Cambiamenti dell'ambiente con conseguenze impreviste
  - Nuove funzionalità di prodotti concorrenti

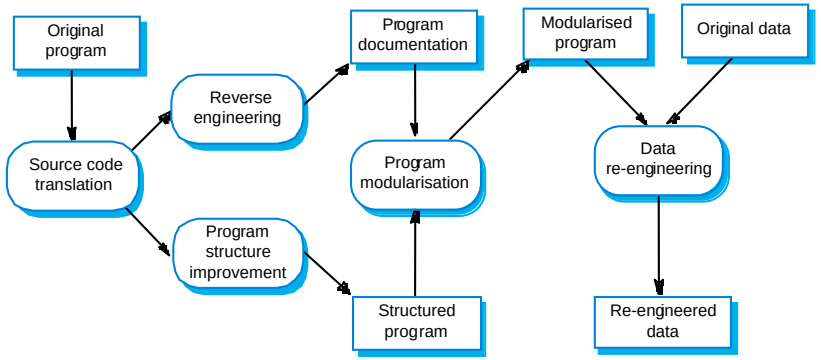
Re-ingegnerizzazione

- Ristrutturazione o riscrittura di un sistema che non ne modifica la funzionalità
- Applicabile quando alcune parti di un sistema richiedono manutenzione frequente
- Comporta investimenti per rendere il sistema più evolvibile. Può richiedere nuova documentazione.
- Vantaggi rispetto allo sviluppo di un nuovo sistema:
  - Riduzione dei rischi
  - Riduzione dei costi

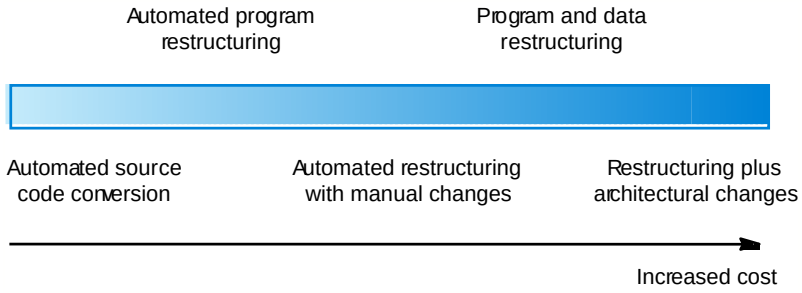
Sviluppo e re-ingegnerizzazione



Processo di re-ingegnerizzazione



Approcci alla re-ingegnerizzazione



Fattori di costo

- Qualità del software esistente
- Disponibilità di strumenti
- Quantità di dati da convertire
- Disponibilità di uno staff esperto e con buona conoscenza del sistema
  - Preferibilmente le stesse persone addette alla manutenzione
  - Problematica per sistemi basati su tecnologie poco usate

Evoluzione dei sistemi ereditati (legacy)

- Strategie di evoluzione di sistemi ereditati:
  - Eliminazione del sistema
  - Manutenzione del sistema
  - Re-ingegnerizzazione del sistema
  - Sostituzione con un nuovo sistema
- La scelta dipende dalla qualità e dal valore economico del sistema

Classificazione e strategie

- Bassa qualità, basso valore:
  - smantellamento
- Bassa qualità, alto valore:
  - re-ingegnerizzazione
- Alta qualità, basso valore:
  - manutenzione finché i costi rimangono bassi, poi smantellamento
- Alta qualità, alto valore:
  - manutenzione

Valutazione del valore economico

- Per mezzo di interviste agli stakeholder del sistema (utenti finali, clienti, manager)
- Aspetti principali:
  - Uso del sistema
  - Processi aziendali supportati
  - Affidabilità del sistema
  - Output del sistema

Valutazione della qualità

- Valutazione del processo di business
  - Quanto è efficace nel conseguire gli obiettivi?
- Valutazione dell'ambiente
  - Quanto è efficace e quanto è costoso da mantenere?
- Valutazione dell'applicazione
  - Qual è la qualità dell'applicazione software?

Valutazione del processo di business

- Domande agli stakeholder
  - Esiste un modello di processo ben definito? Viene seguito?
  - Parti diverse dell'organizzazione usano processi diversi per la stessa funzione?
  - Il processo ha subito adattamenti?
  - Quali sono le relazioni con altri processi di business? Sono necessarie?
  - Il processo è supportato efficacemente dal sistema legacy?

Valutazione dell'ambiente - 1

- Stabilità del fornitore: esiste ancora? Esisterà a lungo? Potrà essere sostituito nella manutenzione?
- Frequenza di fallimenti: hardware e software
- Età: più l'ambiente è datato, meno sarà funzionale e più costoso da tenere in vita
- Prestazioni

Valutazione dell'ambiente - 2

- Requisiti di supporto: quanto supporto è richiesto per hardware e software, e a che costo?
- Costi di manutenzione: per hardware, software e licenze
- Interoperabilità: l'ambiente funziona correttamente con i sistemi attualmente in uso? Necessita di interfacce o emulazioni apposite?

Valutazione dell'applicazione - 1

- Comprensibilità: complessità delle strutture dati, identificatori significativi, etc.
- Documentazione: è aggiornata e coerente?
- Dati: è disponibile un modello dati per il sistema? Ci sono ridondanze o incoerenze nei dati utilizzati?
- Prestazioni: le prestazioni sono adeguate? Prestazioni insoddisfacenti hanno effetti sul lavoro degli utenti?

Valutazione dell'applicazione - 2

- Linguaggio di programmazione: è supportato da ambienti di programmazione moderni? E' facile trovare programmatori?
- Gestione delle configurazioni: le versioni del sistema sono gestite esplicitamente?
- Dati di test: esistono dati per test di regressione?
- Capacità del personale: è sufficiente per una manutenzione efficace del sistema?

Valutazione dell'applicazione - 3

- Misure quantitative:
  - Numero di richieste di modifica: più è alto, minore è la qualità del sistema (e le modifiche la peggioreranno ancora)
  - Numero di interfacce utente: più è alto, più è probabile che ci siano incoerenze
  - Volume di dati scambiato: più è alto, più complesso è il sistema
- Difficili da interpretare

Manutenzione preventiva

- Apportare miglioramenti pianificati al software
- Proposta per prevenire futuri problemi di manutenzione