



Corso di Project Management

La Manutenzione del Software

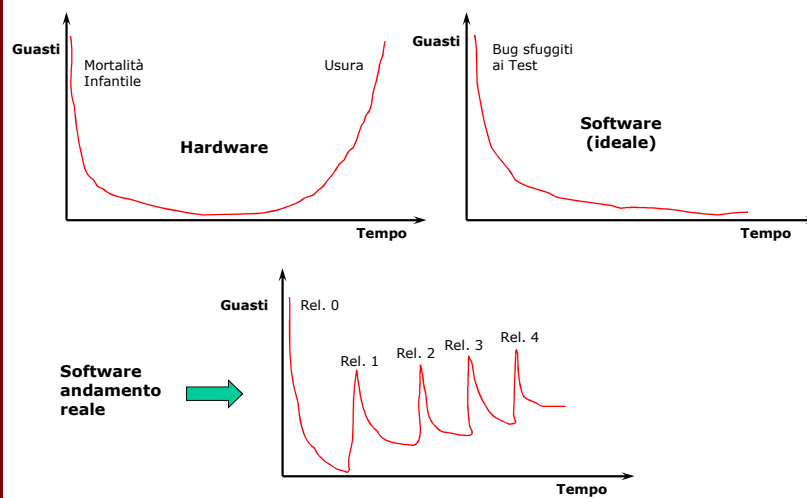
Roberto D'Orsi

Anno Accademico 2013/2014



La manutenzione del software

L'andamento dei guasti nel tempo





La manutenzione del software

I tipi di manutenzione

- **Manutenzione correttiva:** ha l'obiettivo di correggere gli errori di logica o di programmazione che vengono rilevati dopo il rilascio del software. Incide mediamente per il 20-25% sul costo di gestione del SW
- **Manutenzione adeguativa (o adattativa):** riguarda gli adeguamenti del SW a mutamenti dell'ambiente tecnologico o operativo nel quale il SW stesso gira (sistema operativo, DBMS, middleware, hardware,...). Incide mediamente per il 20-25% sul costo di gestione del SW
- **Manutenzione evolutiva (o perfettiva o migliorativa):** ha l'obiettivo di migliorare la Qualità del SW (manutenibilità, performance), di aumentare le funzionalità del prodotto, di consolidare in una nuova Release una serie di correzioni a malfunzionamenti. Incide mediamente, a seconda del prodotto SW, per il 50-60% sul costo di gestione del SW.
Corrisponde, in altri settori industriali al rilancio del prodotto, che viene rivisto e riproposto sul mercato (si pensi al *restyling* di un'auto)
- **Manutenzione Preventiva:** modifiche al prodotto software finalizzate a prevenire potenziali problemi che sono stati identificati. Incide mediamente per meno del 5% sul costo di gestione del SW
Quindi la maggior parte dei costi di manutenzione non è dovuta ad errori di sviluppo ma ai continui cambiamenti imposti dall'ambiente in cui opera (vedi prima legge di Lehman)



La manutenzione del software

Il peso dell'attività di manutenzione rispetto allo sviluppo

Manutenzione >> Nuovi Sviluppi	33,3%
Manutenzione > Nuovi Sviluppi	21,7%
Manutenzione = Nuovi Sviluppi	34,8%
Manutenzione < Nuovi Sviluppi	5,8%
Manutenzione << Nuovi Sviluppi	4,3%

Fonte: AICQ - Comitato Qualità del SW - Quaderno 12

- Nonostante questa evidente prevalenza delle attività di manutenzione rispetto a quelle destinate a nuovi sviluppi, raramente le Aziende si preoccupano di razionalizzare e raffinare il processo di manutenzione e/o di acquisire strumenti specifici, preferendo investire in *tool* destinati allo sviluppo per poi adattarli anche alle esigenze del processo di manutenzione
- Anche sul fronte del controllo dei costi, mentre sui progetti di sviluppo viene fatto (normalmente) un'accurata contabilità di tempi e costi, non sono molte le Aziende che effettuano un controllo rigoroso dei costi di manutenzione del singolo prodotto SW



La manutenzione del software

Difetti, errori, malfunzionamenti

- **Difetto (*Fault*):** elemento del software, oppure parte di testo della documentazione relativa al prodotto software, che non risponde ai requisiti. Un solo difetto può generare n malfunzioni
- **Errore (*Error*):** azione umana che ha generato il difetto
- **Malfunzione, Malfunzionamento (*Failure*):** conseguenza di un difetto che si manifesta durante l'utilizzo del prodotto software, che risulta non conforme, in qualche suo comportamento, ai requisiti specificati dal Cliente

L'obiettivo della fase di test è appunto quello di eliminare la maggior parte possibile di difetti prima che il prodotto software venga utilizzato, in modo da evitare qualsiasi malfunzionamento prevedibile

Il test deve verificare se e in che condizioni:

- Un programma NON FA ciò che dovrebbe fare
- Un programma FA ciò che NON dovrebbe fare

Non dimentichiamo che il testing è una delle principali (anche se non l'unica) attività di *Quality Assurance* di un prodotto software



La manutenzione del software

Gravità delle malfunzioni

- Una malfunzione può essere di tipo funzionale, oppure essere relativa all'interfaccia utente (problema di usabilità), oppure essere di tipo non funzionale (prestazioni, sicurezza, accuratezza, uso di risorse,...)
- Una malfunzione può essere classificata in base alla gravità delle conseguenze sull'utilizzo del prodotto software, esempio:
 - **Malfunzione grave:** l'intero sistema è indisponibile per gli utenti
 - **Malfunzione rilevante:** alcune funzionalità critiche del sistema sono indisponibili per gli utenti
 - **Malfunzione media:** alcune funzionalità non critiche del sistema sono indisponibili per gli utenti, qualche difficoltà nell'utilizzo del prodotto
 - **Malfunzione lieve:** alcune funzionalità non critiche del sistema sono indisponibili per gli utenti, ma non c'è impatto significativo sull'utilizzo del prodotto
- Il livello di gravità di una malfunzione viene utilizzato, in molti contratti di manutenzione SW, come metrica che regola i tempi di intervento (vedi concetto di SLA = *Service Level Agreement*)



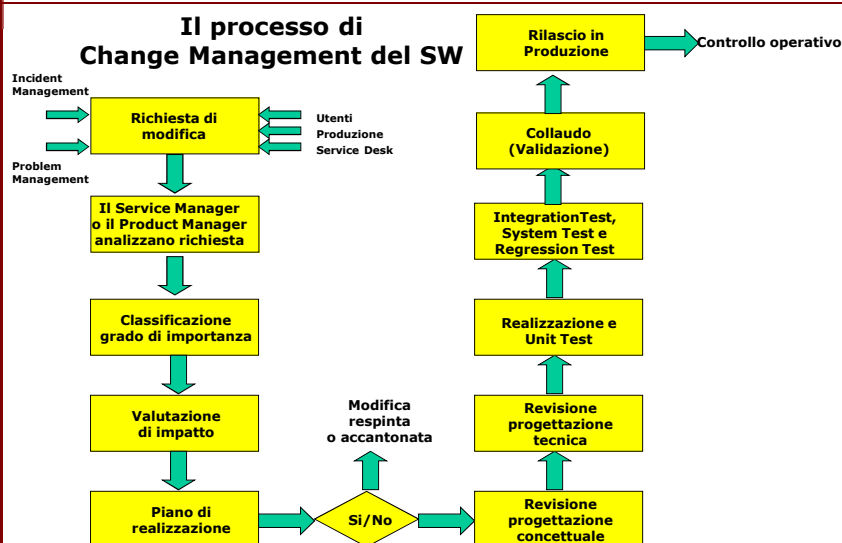
La manutenzione del software

Il processo di data-fixing

- In alcuni casi la malfunzione può generare dati errati nei database e non è detto che ci si accorga immediatamente del problema
- Oltre alla correzione della malfunzione che ha generato l'errore è quindi necessario in questi casi procedere anche ad una bonifica dei dati errati: è necessario identificarli e procedere alla loro correzione, creando (ove possibile) una procedura "one-shot" in grado di forzare il dato corretto
- In alcuni casi il ripristino dei dati può risultare un'operazione lunga e complessa, in quanto si deve recuperare l'input cartaceo che ha generato il dato
- E' evidente che il *data-fixing* è un'operazione molto delicata, in quanto si vanno a forzare dati nel DB in modo più o meno automatico, a seconda della soluzione, ma in ogni caso bypassando tutti i controlli di congruità del dato che vengono normalmente effettuati dall'applicazione



La manutenzione del software





La manutenzione del software

Alcune considerazioni sul Change Management del software

- Una *Request for Change* (R.f.C.) può essere motivata da motivazioni proattive (riduzione di costi, miglioramento del servizio, eliminazione di rischi latenti,..) oppure motivazioni reattive (interruzioni di servizio, richieste utente, adeguamenti a cambiamenti di ambiente,...)
- In entrambi i casi le R.f.C. devono essere adeguatamente controllate per fare in modo di ridurre al minimo sia la probabilità che l'impatto dei possibili rischi legati alla modifica, compreso quello di dover tornare indietro o di dover fare un ulteriore passaggio in produzione per correggere gli errori del primo tentativo
- Tutte le modifiche effettuate devono essere autorizzate, valutate, autorizzate, pianificate in base alla priorità, implementate, testate, documentate, validate
- Nessuna R.f.C. dovrebbe essere autorizzata senza aver preventivamente pianificato quali azioni intraprendere in caso di fallimento del *Change*



La manutenzione del software

Il controllo delle modifiche

Pertanto per ogni richiesta di modifica ad un elemento di configurazione deve rimanere evidenza oggettiva in un apposito Archivio di Configurazione (**C**onfiguration **M**anagement **D**ata**B**ase):

- Chi ha richiesto la modifica
- Quale è il motivo della modifica
- Quali sono i ritorni attesi dalla modifica
- Quali sono i rischi associati
- Esito dell'analisi e valutazione della richiesta di modifica
- Quali sono le risorse necessarie in termini di *skill*
- Chi è responsabile della implementazione
- Data/ora della modifica
- Che cosa è stato modificato
- Tipo di modifica: impatto limitato sul servizio, considerevole impatto sul servizio, modifica in emergenza
- Correlazione con gli altri eventuali elementi che sono stati modificati per risolvere lo stesso problema
- Esito del rapporto di *audit* della configurazione (garanzia della completezza e correttezza funzionale e fisica degli elementi rispetto ai requisiti)



La manutenzione del software

La tracciabilità delle modifiche

Per ogni livello di Versione-Release-Modifica (VRM) deve essere possibile associare ad ogni manutenzione effettuata (correttiva, evolutiva o adeguativa) le motivazioni della modifica e l'elenco degli oggetti SW modificati

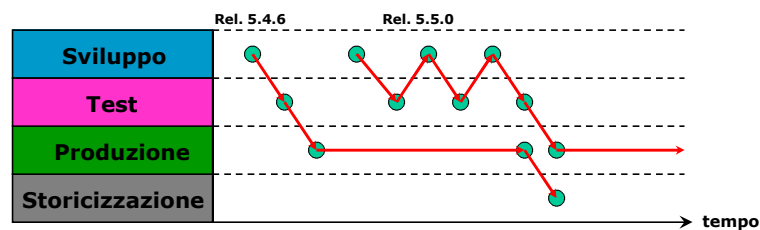
La ricostruibilità delle modifiche

Deve essere possibile in qualsiasi momento poter ricreare una VRM precedente all'ultima rilasciata (n-1) per poter effettuare la diagnosi di eventuali malfunzionamenti e per poter tornare rapidamente alla versione precedente (*back-out*) nel caso di gravi problemi.

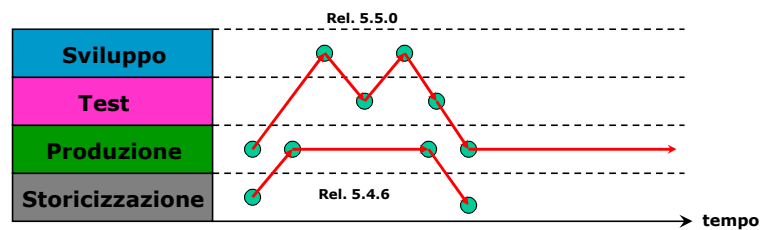
Il ripristino della versione precedente deve essere possibile anche se sono state effettuate congiuntamente modifiche al database: in quest'ultimo caso va predisposta una specifica procedura di ripristino



La manutenzione del software



Il passaggio in produzione di una Release



Il ripristino di una Release precedente (*back-out*)



La manutenzione del software

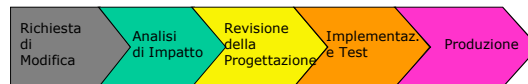
Cosa deve poter consentire la tracciabilità delle modifiche

- La tracciabilità deve poter essere verificata sia partendo dalla registrazione dei test che partendo dalla versione e dallo stato dei moduli sotto configurazione
- Deve essere ricostruibile in qualsiasi momento la correlazione tra moduli gestiti dalla configurazione SW e registrazione delle anomalie, moduli modificati, registrazione delle modifiche, registrazione dell'esecuzione e dell'esito dei test, registrazione del cambiamento di VRM dei moduli dopo l'esito positivo dei test
- Deve essere possibile in qualsiasi momento poter identificare in modo corretto e certo la *baseline* corrente del prodotto, comprendente tutta la documentazione relativa alle modifiche richieste e all'esito dei test



La manutenzione del software

Il ciclo di vita waterfall semplificato della manutenzione correttiva – evolutiva – adeguativa - preventiva





La manutenzione del software

Gli aspetti organizzativi della manutenzione

Struttura dedicata: è applicabile solo in strutture di grandi dimensioni, nelle quali è possibile dimensionare la massa critica di risorse con adeguato *skill* in grado di provvedere alla manutenzione di tutto il parco applicativo, ognuno per la sua parte di competenza

Vantaggi:

- 1) chiara attribuzione di responsabilità
- 2) tempi mediamente rapidi di presa in carico
- 3) alta specializzazione → alta produttività
- 4) maggior facilità di gestire le priorità
- 5) maggiore controllo del processo
- 6) possibilità di contabilizzare in modo preciso i costi
- 7) maggiore facilità di adozione di *tool* specifici

Svantaggi:

- 1) possibile demotivazione del personale addetto
- 2) rischio di un incompleto trasferimento di *know-how* tra personale di sviluppo e di manutenzione



La manutenzione del software

Gli aspetti organizzativi della manutenzione (segue)

Struttura condivisa : è la modalità più diffusa nella maggior parte delle organizzazioni. Ogni risorsa, oltre ad essere assegnata allo sviluppo di nuove applicazioni, ha la responsabilità della manutenzione dei prodotti SW di sua competenza

Vantaggi:

- 1) bilanciamento dinamico del carico di lavoro tra sviluppo e manutenzione in funzione delle priorità
- 2) migliore distribuzione del *know-how*
- 3) maggiore motivazione delle persone
- 4) maggiori opportunità di crescita professionale per tutti (nessuno rimane vincolato solo ai vecchi prodotti)

Svantaggi:

- 1) soprattutto nelle Aziende con un parco applicativo piuttosto vasto, realizzato in casa, obsoleto e pieno di problemi c'è il rischio di non riuscire mai a far decollare i nuovi progetti nuovi per mancanza risorse
- 2) rischio di sottovalutare la necessità di impostare un sistema di controllo del processo di manutenzione con indicatori del livello di servizio e della Qualità erogata



La manutenzione del software

Gli aspetti organizzativi della manutenzione

Outsourcing : è una modalità che si sta diffondendo in molte Aziende. La manutenzione dell'intero parco applicativo (o di una sua parte) viene affidata ad un fornitore esterno

Vantaggi: 1) le risorse interne possono essere dedicate ad attività di sviluppo di nuovi progetti o comunque di più alto livello

2) si risolvono molti problemi di obsolescenza tecnica delle risorse interne

3) possibile diminuzione dei costi di gestione

Svantaggi: 1) è una soluzione da cui è difficilissimo tornare indietro (*insourcing*)

2) gli aspetti contrattuali sono delicatissimi, occorre concordare in modo molto accurato i livelli di servizio e tutte le possibili clausole, eventuali penali

3) si rischia di rimanere in balia del fornitore

4) il rischio di un cattivo trasferimento di *know-how* tra addetti allo sviluppo (interni) e *outsourcer* è alto



La manutenzione del software

Alcuni possibili KPI del Processo di Software Maintenance

- Tempo medio per la presa in carico del problema
- Tempo medio per l'analisi del problema
- Tempo medio per il *test*, compreso quello di regressione
- Tempo medio per la risoluzione complessiva del problema (MTTR = Medium Time To Repair)
- Dimensione delle modifiche apportate (LOC = n° Lines Of Code)
- Indice di gestione del backlog = numero problemi chiusi nella settimana/numero problemi emersi nella settimana
- Indice di difettosità delle modifiche = modifiche difettose/modifiche installate
- Indice di difettosità di un modulo software = tempo medio tra due malfunzioni successive (MTBF = Medium Time Between Failures)
- Percentuale dei Change che hanno soddisfatto la richiesta
- Percentuale dei Change non autorizzati
- Percentuale dei Change in emergenza
- Percentuale dei Change falliti che hanno richiesto un *back-out*
- Numero di incidenti riconducibili ad un Change
- Rapporto tra i benefici economici di un Change (quando valutabili) rapportati al costo del processo di Change