

## La manutenzione del software

---

- Come definirla ?
- Come si colloca nel ciclo di vita del software ?
- Quali problemi risolve ?
- Quale è il suo impatto economico ?
- In che cosa consiste il processo di manutenzione ?
- Chi deve effettuarla?
- Come organizzare il gruppo di manutenzione?

## Come definirla

---

Changes that have to be made to computer programs after have been delivered to the customer or user (Martin e McClure 1983)

The performance of those activities required to keep a software system operational and responsive after it is accepted and placed into production (Federal Information Processing Std. 1994)

Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment (IEEE 1219 1993)

## Come si colloca nel ciclo di vita del software

---

Tutte le definizioni concordano sul fatto che:

- la manutenzione è una fase del CVS;
- non esiste attività di manutenzione prima che un prodotto software divenga operativo

Naturalmente le relazioni esistenti tra la fase di manutenzione e le altre fasi del CVS variano in funzione del modello di CVS che viene adottato

## Quali problemi risolve

---

Tutti i problemi connessi alle prime due leggi di Lehman:

La Manutenzione è *inevitabile*

I requisiti di un sistema software sono in continua evoluzione in risposta ai cambiamenti del dominio applicativo, delle tecnologie, del obiettivi degli utenti, etc.

I sistemi software devono essere mantenuti affinché essi rimangano utili.

Lehman nel 1985 ...

## Leggi di Lehman - 1

---

- a) **Law of Continuing Change** che stabilisce che un sistema deve necessariamente cambiare ed evolvere con continuità per mantenere intatto il suo livello di utilità, stante i continui cambiamenti che avvengono nell'ambiente in cui il sistema opera
- b) **Law of Increasing Complexity** che stabilisce che la struttura di un sistema che evolve si deteriora man mano che cambia e che risorse aggiuntive devono essere impegnate per preservarne la semantica e semplificarne la struttura.

## Leggi di Lehman - 2

---

- c) **Law of large program evolution** che stabilisce che l'evoluzione di un Programma è un processo auto-regolante. Gli attributi del Sistema come grandezza, tempo tra le varie versioni, ed il numero di errori rilevati sono approssimativamente invarianti per ciascuna versione del sistema
- d) **Law of organizational stability** che stabilisce che durante il ciclo di vita di un programma, la sua velocità di sviluppo è approssimativamente costante ed indipendente dalle risorse dedicate allo sviluppo del sistema
- e) **Law of conservation of familiarity** che stabilisce che durante il ciclo di vita di un sistema i cambiamenti in ciascuna versione è approssimativamente costante

## Perché mantenere un sistema software

---

- Correzione di errori del codice e del disegno
- Interfacciamento con altri sistemi
- Miglioramenti nelle prestazioni o di altri attributi di qualità quale la stessa manutenibilità
- Apportare modifiche necessarie a funzionalità, files e basi di dati
- Consentire l'impiego di differenti piattaforme hardware, software di base e di ambiente

## Tipi di manutenzione - 1

---

Secondo Swanson i tipi di manutenzione sono:

**correttiva:** modifiche per la rimozione di errori residui

**adattativa:** modifiche legate all'ambiente (s.o. , piattaforme software in genere, ecc.) in cui il sistema software deve operare

**perfettiva:** tutte i cambiamenti apportati al sistema per venire incontro a necessità dell'utente

## Tipi di manutenzione - 2

Lo standard IEEE 1219 - 1993 definisce i seguenti tipi:

**correttiva:** reactive modification of a software product performed after delivery to correct discovered problem

**adattativa:** modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment

**perfettiva:** modification of a software product performed after delivery to improve performance or maintainability

**preventiva:** modification of a software product performed after delivery to prevent problems before they occur

**di emergenza:** unscheduled corrective maintenance performed to keep a system operational

## Come si ripartiscono i costi della manutenzione

Con riferimento ai tipi di manutenzione secondo Swanson si può assumere:

- manutenzione correttiva: 20%
- manutenzione adattativa: 25%
- manutenzione perfettiva: 55%

La maggioranza dei costi di manutenzione non sono quindi attribuibili a errori di sviluppo ma ai continui cambiamenti imposti dall'ambiente in cui un sistema software opera (cfr. 1a legge di Lehman)

## Quale è l'impatto economico

Tutti gli studi effettuati concordano nel ritenere che :

- la maggior parte del costo del CVS è attribuibile alla manutenzione
- percentualmente la sua incidenza sul costo complessivo è crescente nel tempo e allo stato tale percentuale è stimabile pari a 80% (cfr. 2a legge di Lehman)
- la maggior parte degli addetti al CVS lavora in manutenzione

## I costi della manutenzione - 1

- **Indipendenza dei moduli.** Dovrebbe essere possibile modificare un modulo limitando le modifiche agli altri.
- **Linguaggio di programmazione.** I sistemi software sviluppati con linguaggi di programmazione di alto livello sono più semplici da mantenere.
- **Stile di programmazione.** La programmazione strutturata facilita la manutenzione.
- **Validation e testing.** Una buona attività di validation and testing contribuisce a limitare gli interventi di manutenzione correttiva.

## I costi della manutenzione – 2

- **Documentazione.** Una documentazione esaustiva favorisce la comprensione dei sistemi.
- **Configuration management.** Un buon configuration management assicura un'efficace tracciabilità tra i diversi artefatti di un sistema.
- **Dominio applicativo.** La manutenzione di sistemi che operano in domini noti e maturi risulta meno complessa.
- **Staff.** I costi di manutenzione possono essere ridotti limitando il turn-over nell'ambito dello staff di manutenzione.

## L'ingegneria del software esistente - 1

È la naturale risposta alla richiesta di teorie, metodi, tecniche e tecnologie a supporto della manutenzione adattativa e perfetta dei sistemi software ereditati (legacy systems)

Tale richiesta è sempre crescente in quanto:

- per le aziende produttrici di software e per gli utenti il portafoglio software ed i sistemi in esercizio rappresentano un patrimonio ed un capitale alla cui rinuncia è spesso impossibile pensare
- le innovazioni introdotte nella produzione di nuovi sistemi dalla ingegneria del software producono un aumento dell'entropia del software in esercizio con conseguente crescita della domanda di manutenzione

## L'ingegneria del software esistente - 2

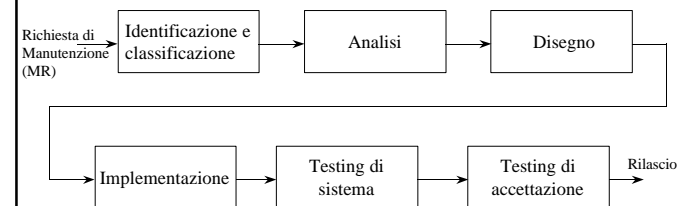
Ha peraltro uno spettro di interessi che travalica la manutenzione per comprendere il riuso di un qualsiasi prodotto del ciclo di produzione e della relativa conoscenza.

Alcuni settori:

|                 |                         |                       |
|-----------------|-------------------------|-----------------------|
| restructuring   | reverse engineering     | conversion            |
| design recovery | reengineering           | program comprehension |
| renewal         | business rules recovery | reuse-reengineering   |
| recycling       | impact analysis         | conceptual data model |
| down-sizing     | restoration             | bridge technology     |

## Lo Standard IEEE 1219 (1993)

Definisce un modello di processo di manutenzione di tipo iterativo articolato in sette fasi per gestire ed eseguire attività di manutenzione.



## Identificazione e classificazione - 1

In questa fase una richiesta di modifica (MR) è identificata, classificata, associata ad una priorità e valutata

### **Input**

MR avanzata da: utenti, programmatori e/o manager

### **Attività di processo della fase**

- codifica della MR e sua classificazione
- analisi preliminare per decidere se accettarla, respingerla o valutarla ulteriormente
- stima preliminare della sua dimensione/ampiezza
- determinazione della sua priorità
- assegnazione della MR a un blocco di modifiche già previste

## Identificazione e classificazione – 2

### **Attività di controllo della fase**

- memorizzazione della MR e dei risultati delle attività svolte in una base dati

### **Output**

I risultati delle attività svolte in termini di:

- definizione del problema o dei nuovi requisiti
- valutazione del problema o dei nuovi requisiti
- classificazione della MR e sua priorità iniziale
- verifica dei dati per le MR di tipo correttivo
- stima iniziale delle risorse necessarie

## Analisi – 1

Vengono valutati fattibilità e scopo delle MR (analisi di fattibilità) e sviluppato un piano preliminare per il disegno, implementazione test e rilascio delle modifiche da apportare (analisi di dettaglio)

### **Input**

- l'output della fase precedente
- la documentazione del sistema software (documentazione di sistema e di progetto)

## Analisi – 2

### **Attività di processo dalla fase**

#### A) analisi di fattibilità per valutare:

- impatto
- soluzioni alternative
- analisi dei requisiti di conversione
- fattori umani
- costi a breve e a lungo termine
- benefici ottenuti con le modifiche

#### B) analisi dettagliata per:

- definire i requisiti richiesti all'azienda dalla modifica
- individuare gli elementi da modificare
- definire una strategia di testing
- pianificare la realizzazione dell'intervento

### **Analisi – 3**

---

#### *Attività di controllo della fase*

- identificazione della versione più significativa della documentazione del sistema software
- revisione delle modifiche proposte per validarne la fattibilità tecnico/economica e la correttezza
- identificazione dei problemi di sicurezza
- valutazione dell'integrazione delle modifiche nel software esistente
- verifica che la documentazione disponibile è aggiornata
- verifica che esiste una strategia di testing e che questa è idonea per le modifiche da apportare
- scelta delle modifiche da implementare nella nuova release

### **Analisi – 4**

---

#### *Output*

Report di:

- fattibilità della MR
- analisi dettagliata
- requisiti da aggiornare
- lista preliminare delle modifiche
- strategia di testing da adottare
- pianificazione delle attività di disegno ed implementazione

### **Analisi – 5**

---

Al termine della fase i risultati dell'analisi dettagliata dovranno consentire alla struttura aziendale preposta ad autorizzare le modifiche (struttura di gestione della configurazione) di decidere:

- se la richiesta debba essere soddisfatta o meno
- in quale futura release dovranno essere inserite le modifiche

Naturalmente:

- gli interventi di manutenzione correttiva possono essere gestiti diversamente
- deve essere opportunamente definita la procedura di rilascio di nuove release dovute ad interventi di manutenzione di emergenza

### **Disegno - 1**

---

Vengono raccolte tutte le informazioni disponibili (documentazione del sistema software, risultati della fase di analisi, codice, ecc.)

I documenti, peraltro già presi in esame durante la precedente fase di analisi, vengono rivisitati nell'ottica di progettare le modifiche.

Le modifiche vengono progettate e sono aggiornati i documenti di analisi dettagliata

## Disegno - 2

---

### *Input*

- output della fase precedente
- documentazione di sistema e di progetto
- codice sorgente, commenti e basi di dati

### *Attività di processo della fase*

- identificazione dei moduli affetti dalle modifiche
- modifica della documentazione di disegno dei moduli software
- definizione dei test case per le modifiche da apportare
- identificazione/creazione dei test case di regressione
- lista aggiornata delle modifiche

## Disegno - 3

---

### *Attività di controllo della fase*

- ispezione dei documenti di disegno
- verifica che i nuovi requisiti e il nuovo disegno è conforme a quanto autorizzato
- verifica della inclusione della nuova documentazione in quella preesistente
- verifica che la documentazione di testing è stata aggiornata
- verifica della tracciabilità dai nuovi requisiti al nuovo disegno

## Disegno - 4

---

### *Output*

- lista delle modifiche rivisitata
- documentazione di disegno aggiornata
- piani di test aggiornati
- analisi di dettaglio aggiornata
- requisiti verificati
- piano dell'implementazione rivisitato
- lista dei vincoli e dei rischi di fallimento

## Implementazione - 1

---

Le modifiche vengono effettuate sul codice.

Viene inoltre effettuato:

- il testing di unità e di integrazione
- l'aggiornamento di tutta la documentazione inclusi i documenti di disegno, di testing, la documentazione utente e di training

È altamente consigliabile che tutti gli oggetti costituenti il sistema software siano gestiti da un gestore automatico di configurazione che garantisce sia l'indipendenza delle varie versioni del sistema sia la consistenza di tutti gli oggetti afferenti alla stessa versione

## Implementazione - 2

---

### *Input*

- output della fase precedente
- documentazione di sistema e di progetto
- codice sorgente, commenti e basi di dati

### *Attività di processo della fase*

- modifiche sul codice
- testing di unità
- integrazione delle modifiche nel codice preesistente
- testing di integrazione
- rivisitazione dell'analisi dei rischi di fallimento

## Implementazione - 3

---

### *Attività di controllo della fase*

- ispezione del codice
- verifica che il testing di unità e di integrazione sia stato fatto e documentato
- verifica che piani di test, test case e procedure di testing siano stati aggiornati
- verifica che il nuovo software sia stato posto sotto il controllo del gestore della configurazione software
- verifica che la documentazione tecnica e di training è stata aggiornata
- verifica della tracciabilità dal disegno al codice

## Implementazione - 4

---

### *Output*

- software aggiornato
- documentazione di disegno aggiornata
- piani di test aggiornati
- documentazione utente aggiornata
- documentazione di training aggiornata

## Testing di sistema - 1

---

Verifica che il sistema soddisfi i requisiti di cui alle modifiche e i requisiti originari.

È uno dei punti critici del processo di manutenzione anche perché le teorie e le tecniche di testing sono orientate al testing per lo sviluppo piuttosto che verso il testing in manutenzione

Richiede l'esecuzione del testing di regressione che è un testing specifico della manutenzione. A tal fine è essenziale che sia disponibile un database di test case che qualora manchi deve essere necessariamente creato e tenuto aggiornato

## Testing di sistema - 2

---

### *Input*

- output della fase precedente
- documentazione di testing di sistema (piani di test, test case, procedure di test)
- manuali utente documentazione di disegno

### *Attività di processo della fase*

- test funzionale di sistema
- testing di interfaccia
- testing di regressione
- preparazione al testing di accettazione

## Testing - 3

---

### *Attività di controllo della fase*

Sarebbe opportuno che le attività di testing fossero condotte da una "terza parte".

Prima che la fase termini è necessario definire lo status, cioè stabilire la misura secondo cui i criteri di testing sono stati soddisfatti.

### *Output*

- sistema testato e completamente integrato;
- test report;
- review dei test report;

## Testing di accettazione

---

È il classico testing di accettazione sviluppato dal cliente, utente o da una terza parte.

Consiste nell'analisi di report di risultati di test e nell'audit delle funzionalità del sistema di cui si accetta la nuova release.

Particolare cura deve essere posta in questa fase nel verificare che tutta la documentazione, sia di sistema che di utente, sia stata correttamente aggiornata.

Un sistema la cui documentazione non è aggiornata non deve superare la fase di testing di accettazione e quindi non è pronto per la messa in esercizio

## Rilascio

---

È la fase in cui il fornitore installa presso il cliente la nuova versione e la messa in esercizio

Sono da effettuare in questa fase:

- una puntuale verifica della consegna di tutti gli oggetti costituenti la nuova versione
- il back-up della vecchia versione per archiviazione
- il training dell'utente sulla nuova versione rilasciata

## Lo Standard ISO/IEC 12207 (1995) - 1

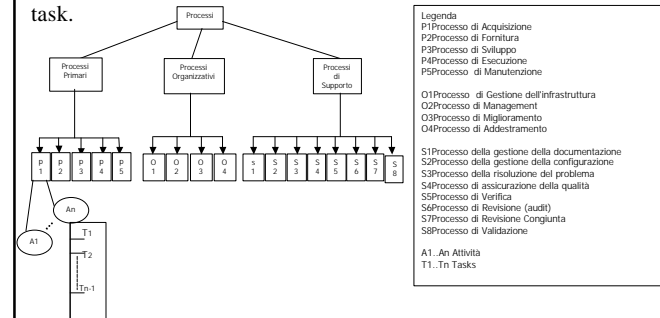
A differenza dello IEEE 1219 tratta una problematica molto più ampia della manutenzione del software.

Obiettivo dello Standard è quello di fornire un quadro di riferimento comune per tutti i processi del ciclo di vita del software con una ben definita terminologia.

Lo standard individua processi, attività e task che devono essere sviluppati durante l'acquisizione di un sistema e dei relativi servizi e durante la fornitura, lo sviluppo, l'esercizio e la manutenzione di prodotti software.

## Lo Standard ISO/IEC 12207 (1995) - 2

ISO 12207 prevede 3 categorie di processi: Processi Primari, Processi Organizzativi e Processi di supporto.  
Ogni processo è a sua volta suddiviso in attività e ciascuna attività in task.



## Lo Standard ISO/IEC 12207 (1995) - 3

I processi Primari gestiscono le funzioni principali del ciclo di vita.

I processi di Organizzazione sono utilizzati per stabilire, controllare e migliorare un processo del ciclo di vita.

I processi di Supporto sono di supporto ad un altro processo nell'eseguire una specifica funzione.

Lo standard definisce per ciascun processo la lista delle attività e dei task; inoltre, fissa gli obiettivi e le responsabilità.

Ai processi di cui prima può aggiungersi un processo di Tailoring attraverso cui è possibile adattare lo standard a particolari progetti (cancellando attività non significative o inserendo nuove attività e compiti ove necessario).

## Il processo di manutenzione

Definisce le attività del manutentore ossia dell'organizzazione che fornisce il servizio di manutenzione di un prodotto software: cioè che gestisce le modifiche del prodotto per tenerlo attuale e in esercizio ottimale preservandone l'integrità

Il processo include tra l'altro la pianificazione delle attività di manutenzione, la migrazione e il ritiro del software.

Il processo di pianificazione precede naturalmente il processo di modifica e quindi deve necessariamente iniziare prima del rilascio del prodotto software. Ne deriva che il processo di manutenzione non è sequenziale ma in parte parallelo al processo di sviluppo

## La manutenzione: una nuova definizione

La manutenzione del software è la totalità delle attività richieste per fornire un supporto economicamente vantaggioso ad un sistema software

Le attività sono sviluppate sia prima che dopo il rilascio.

Le attività che precedono il rilascio includono la pianificazione delle operazioni dopo il rilascio, le modalità con cui viene fornito il supporto e la logistica.

Le attività dopo il rilascio includono i processi di modifica, il training e di help desk. (Pigoski 1996)

## Le attività del processo di manutenzione

Secondo lo Standard il processo di manutenzione si articola nelle seguenti attività:

- *Process Implementation*
- *Problem and Modification Analysis*
- *Modification Implementation*
- *Maintenance Review and Acceptance*
- *Migration*
- *Software Retirement*

## Process Implementation

È l'attività di definizione del processo di manutenzione e delle relative procedure.

Consiste nei seguenti task:

- *Maintenance Planning*
- *Modification Request*
- *Configuration Management*

## Maintenance Planning

Il manutentore deve sviluppare, documentare ed eseguire piano e procedure per condurre le attività ed i compiti del processo di manutenzione.

Possibili subtask:

- definire il concetto di manutenzione ed il suo scopo,
- analizzare i possibili manutentori ed identificare il manutentore,
- valutare le risorse necessarie e stimare i costi,
- validare la manutenibilità del sistema,
- analizzare la transizione da sviluppo a manutenzione,
- individuare il processo di manutenzione da usare e documentarlo sotto forma di procedure operative

## Modification Request

Il manutentore deve stabilire le procedure per ricevere, registrare e tracciare i report sul problema e le richieste di modifica dagli utenti e fornire feedback agli utenti

Possibili subtask:

- definire uno schema di identificazione per le richieste di modifica, del tipo di richiesta e della sua priorità
- definire procedure di analisi del trend delle richieste
- definire le procedure che l'utente deve seguire per formulare una richiesta di modifica
- definire come fornire all'utente un feedback iniziale e i feedback durante il processo della richiesta
- definire come immettere la richiesta nel database del sistema

## Software Configuration Management

Il manutentore deve implementare (o stabilire l'interfaccia organizzativa con) il Configuration Management Process per gestire le modifiche al sistema esistente

Possibili subtask:

- pianificare il configuration management e sviluppare le relative procedure
- individuare una versione formalmente approvata del sistema software da mantenere (codice e documenti relativi)
- definire la base di dati che contiene informazioni relative allo stato della configurazione e i relativi report

## Problem and Modification Analysis

È l'attività di analisi di una richiesta di modifica. Consiste nei seguenti task:

- **Verification:** Il manutentore deve verificare il problema
- **Analysis:** Il manutentore deve analizzare il report del problema o la richiesta di modifica per valutare il suo impatto sull'organizzazione, sul sistema esistente e i sistemi che sono interfacciati
- **Alternate Solutions:** Il manutentore deve sviluppare le opzioni per implementare la modifica
- **Documentation:** Il manutentore deve documentare il problema/richiesta di modifica, i risultati dell'analisi e le opzioni di implementazione
- **Approval:** Il manutentore deve ottenere l'approvazione per l'opzione di modifica selezionata come specificato nel contratto

## Modification Implementation

È l'attività di implementazione di una richiesta di modifica. Consiste nei seguenti task:

- **Items to be modified:** Il manutentore deve condurre l'analisi e determinare quale documentazione, unità software, e versione ha bisogno di essere modificata
- **Development process:** Il manutentore deve iniziare il processo di sviluppo per implementare la modifica
- **Testing:** I criteri di test e valutazione per testare e valutare le parti modificate e quelle non modificate del sistema devono essere definite. Deve essere assicurata l'implementazione completa e corretta dei requisiti nuovi e modificati; deve essere anche verificato che i requisiti originali non modificati non sono impattati dall'intervento

## Maintenance Review and Acceptance

---

È l'attività di verifica finale e di accettazione. Consiste nei seguenti task:

- **Review:** Il manutentore deve eseguire la revisione con l'organizzazione che autorizza la modifica per determinare l'integrità del sistema modificato
- **Approval:** Il manutentore deve ottenere l'approvazione per il completamento soddisfacente della modifica come specificato nel contratto

## Migration - 1

---

È l'attività di migrazione del sistema software in un nuovo ambiente operativo. Consiste nei seguenti task:

- **Standard:** Il manutentore deve assicurare che un prodotto software o i dati prodotti o modificati durante la migrazione rispettino lo Standard ISO/IEC
- **Migration Plan:** Un piano di migrazione deve essere sviluppato, documentato ed eseguito. La pianificazione delle attività devono includere gli utenti. I termini inclusi nel piano devono contenere:
  - Analisi dei requisiti e definizione di migrazione
  - Sviluppo di tool di migrazione
  - Conversione del prodotto e dei dati software
  - Esecuzione e verifica della migrazione
  - Supporto per i vecchi ambienti nel futuro

## Migration - 2

---

- **Notification of Intent:** Agli utenti deve essere fornita la notifica del piano e delle attività di migrazione. La notifica deve includere:
  - Motivazione del perché il vecchio ambiente non è più supportato
  - Descrizione del nuovo ambiente con la data della sua disponibilità
  - Descrizione di altre opzioni di supporto disponibili, se c'è ne sono, una volta che il supporto per il vecchio ambiente è stato rimosso
- **Parallel Operation:** L'operatività parallela del vecchio e del nuovo ambiente può essere garantita per un breve periodo durante la transizione verso il nuovo ambiente. Durante questo periodo, è necessario fornire del training come specificato nel contratto

## Migration - 3

---

- **Notification of Migration:** Quando la migrazione schedulata è completata, la notifica deve essere inviata a tutti gli interessati. Tutta la documentazione, i logs ed il codice associati al vecchio ambiente devono essere archiviati
- **Impact analysis:** Deve essere eseguita una revisione post-operazione per valutare l'impatto dei cambiamenti verso il nuovo ambiente. I risultati della revisione devono essere inviati ad apposite autorità per raccogliere informazioni, suggerimenti ed azioni
- **Archives:** I dati usati da o associati a dati del vecchio ambiente devono essere accessibili in accordo con i requisiti di contratto per la protezione dei dati e la verifica applicabile ai dati

## Software retirement - 1

È la attività di ritiro del software da dismettere. Consiste nei seguenti task:

- **Retirement Plan:** Deve essere sviluppato e documentato un piano di ritiro per rimuovere il supporto attivo dalla organizzazione di manutenzione. Le attività di pianificazione devono considerare gli utenti. Il piano deve contenere i seguenti item:
  - Interruzione di un supporto pieno o parziale dopo un certo periodo di tempo
  - Archiviazione del prodotto software e della documentazione associata
  - Responsabilità per un futuro supporto residuo
  - Transizione al nuovo prodotto software, se applicabile
  - Accessibilità alle copie ed ai dati archiviati

## Software retirement - 2

- **Notification of intent:** È necessario notificare agli utenti i piani e le attività di ritiro. La notifica deve includere:
  - Descrizione della sostituzione o dell'aggiornamento con le date di disponibilità
  - Motivazioni del perché il prodotto software non è più supportato
  - Descrizione di altre opzioni di supporto disponibili una volta che il supporto è stato rimosso
- **Parallel operations:** operazioni parallele di ritiro del vecchio sistema e di installazione del nuovo prodotto dovrebbero essere eseguite in parallelo per una più "indolore" transizione al nuovo sistema

## Software retirement - 3

- **Notification of retirement:** Il completamento del ritiro schedulato deve essere notificato a tutti gli interessati. Tutta la documentazione, i logs ed il codice associati allo sviluppo devono essere archiviati, quando è il caso
- **Archives:** I dati usati o associati al prodotto software ritirato devono essere accessibili in accordo con i requisiti del contratto per la loro protezione e la loro auspicabile verifica

## Chi deve effettuarla - 1

### L'organizzazione che ha sviluppato il prodotto

Vantaggi:

- chi ha sviluppato ha la migliore conoscenza del prodotto
- non vi è necessità di disporre di una documentazione estremamente dettagliata
- non è necessario creare collegamenti tra i manutentori e sviluppatori
- l'utente ha una unica controparte
- il personale dell'organizzazione che ha sviluppato il prodotto ha la possibilità di diversificare la propria attività

## Chi deve effettuarla - 2

---

### L'organizzazione che ha sviluppato il prodotto

#### Svantaggi:

- la documentazione del prodotto non sarà mai realmente terminata
- gli sviluppatori spesso dedicano molto tempo per perfezionare il sistema
- spesso coloro che hanno sviluppato vengono dirottati su progetti a più alta priorità
- il turnover può costringere a far mantenere il sistema da personale che non possiede un know-how adeguato
- un eccessivo lavoro di manutenzione può creare insoddisfazione tra gli sviluppatori

## Chi deve effettuarla - 3

---

### Una organizzazione che effettua manutenzione conto terzi

#### Vantaggi:

- verrà prodotta una migliore documentazione
- saranno messe a punto procedure formali per gestire il turnover
- vi potrà essere una obiettiva conoscenza dei punti deboli e forti del sistema
- saranno messe a punto procedure per realizzare gli interventi
- il personale di una organizzazione che effettua manutenzione conto terzi non ritiene una diminuzione il mantenere e quindi lavora meglio

## Chi deve effettuarla - 4

---

### Una organizzazione che effettua manutenzione conto terzi

#### Svantaggi:

- la transizione tra sviluppatori e manutentori può essere lenta
- può richiedere un notevole periodo di training
- l'utente finale può andare incontro a carenze di supporto
- vi possono essere disservizi nella manutenzione del software