

## APPENDICE 1 AL CAPITOLATO TECNICO

### GARA PER LA FORNITURA DI PRODOTTI COMPUTER ASSOCIATES E RELATIVI SERVIZI PER IL SISTEMA INFORMATIVO SOGEI - ID 1543



## INDICE

1. INTRODUZIONE	3
2. ARCHITETTURA DI PRODOTTO E AMBIENTE OPERATIVO	3
2.1. ARCHITETTURA DI PRODOTTO	3
2.2. AMBIENTE OPERATIVO	4
3. CICLO DI VITA DEL SOFTWARE	5
3.1. MODELLO DI RIFERIMENTO	5
3.2. TERMINOLOGIA	6
3.3. PROCESSI OPERATIVI	8
3.4. FLUSSO ELABORATIVO	10
4. GESTIONE DEL REPOSITORY	11
4.1. TIPI DI FILE	11
5. PERSONALIZZAZIONI	11
5.1. STATO LOGICO SVILUPPO	11
5.2. STATO LOGICO COLLAUDO	12
5.3. STATO LOGICO PRODUZIONE	12
6. SINCRONIZZAZIONE AMBIENTI UNIX REMOTI	13
6.1. SINCRONIZZAZIONE STIPENDI	14
6.2. SINCRONIZZAZIONE ANAGRAFICA UNICA	14



## 1. INTRODUZIONE

Il presente documento descrive la soluzione CA Software Change Manager versione 12.5, adottata dal Ministero Economia e Finanze per la gestione dei processi di Change e Configuration Management in ambienti distribuiti, per quanto concerne le sue componenti architetturali, le linee guida funzionali del prodotto e le personalizzazioni adottate nei processi di gestione del ciclo di vita del software dei vari progetti applicativi.

## 2. ARCHITETTURA DI PRODOTTO E AMBIENTE OPERATIVO

### 2.1. ARCHITETTURA DI PRODOTTO

CA Software Change Manager è un prodotto client/server, dotato anche di una componente per l'accesso in modalità web, pensato per ambienti distribuiti, che automatizza il processo di gestione delle configurazioni e dei cambiamenti di ambienti applicativi.

Automatizzare la gestione della configurazione e dei cambiamenti di un ambiente applicativo significa:

- Creare un processo definito e ripetibile.
- Definire un ciclo di vita flessibile.
- Notificare, i cambiamenti e le approvazioni.
- Avere le informazioni relative allo stato del progetto, le metriche, le release software, ecc.

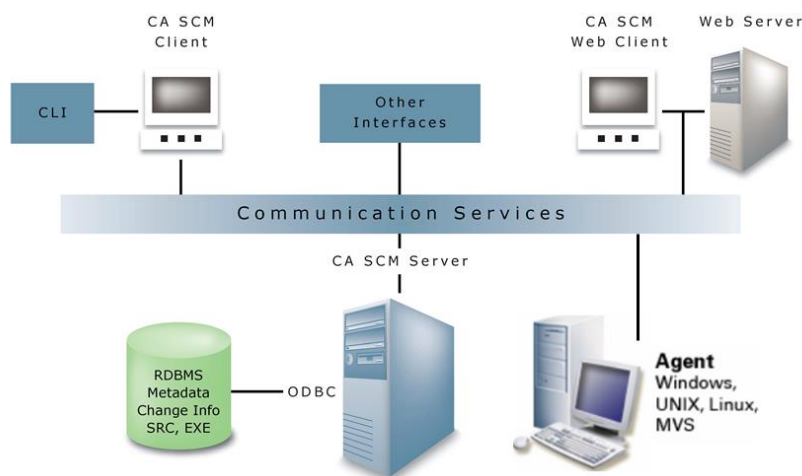


Fig 1. CA Software Change Manager – Schema Architettuale

Nella figura precedente è schematizzata l'architettura di prodotto sulla base delle sue componenti fondamentali, dettagliate di seguito.

**CA SCM Server:** E' il componente che gestisce il repository degli oggetti e che governa ed esegue, sulla base delle richieste provenienti dai clienti, tutti i processi che regolano il flusso degli oggetti tra i vari stati logici (checkin items, checkout concorrente, promote, demote, etc.).



**CA SCM WEB Server:** E'la componente web del prodotto, per consentirne l'utilizzo da remoto senza la necessità di installazione di un CA SCM Client sulla postazione utente.

**DB Server:** Immagazzina gli items di configurazione software ed i dati correlati (packages, utenze, profili di accesso, etc.) in un database relazionale e comunica con CA SCM Server attraverso una connessione ODBC.

**CA SCM Agent:** E' il componente che consente di sincronizzare il repository di CA Software Change Manager con l'ambiente Unix, in modo che a fronte di operazioni di promote di un oggetto in collaudo e produzione, tale oggetto sia copiato anche sulla corrispondente directory dell'ambiente Unix.

**CA SCM Client:** E il componente applicativo che gestisce l'interfaccia utente. Riceve dei comandi di input, li trasmette al server affinché vengano processati e ripresenta all'utente attraverso la Graphical User Interface i risultati dei processi eseguiti sul server. E' installato sui client degli utenti che utilizzeranno il prodotto ed è composto a sua volta dai due componenti **Administrator** (per le attività di gestione e amministrazione, ) e **Workbench** (per le operazioni di check-in, check-out, promote, etc.).

## **2.2. AMBIENTE OPERATIVO**

Di seguito il dettaglio dei server utilizzati in ambito Ministero Economia e Finanze.

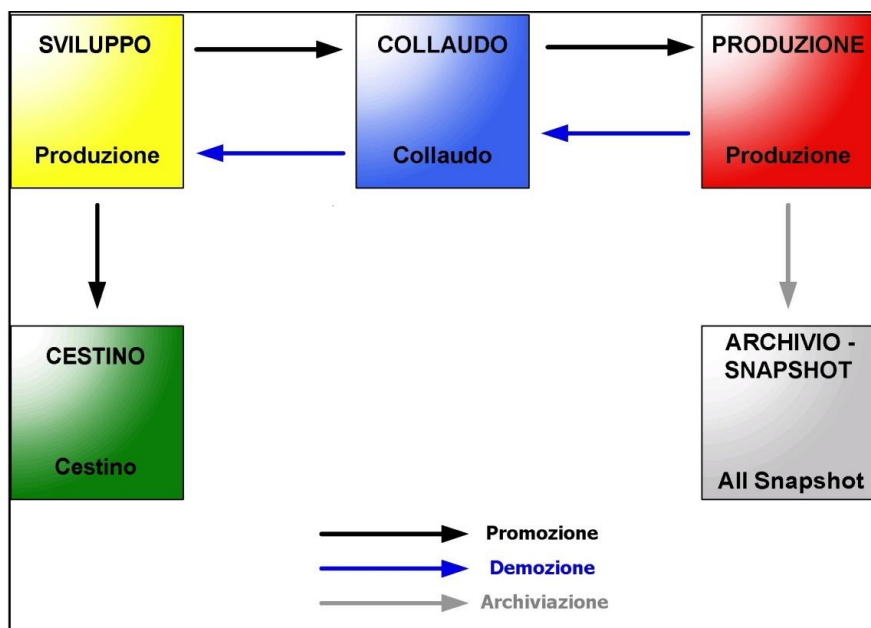
<b><u>CA SCM SERVER</u></b>	O.S : RedHat Enterprise Linux Server rel 5.4
<b><u>DB SERVER</u></b>	DBMS : Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bit
<b><u>CA SCM WEB SERVER</u></b>	O.S : Red Hat Enterprise Linux Server release 6.5
<b><u>CA SCM AGENT</u></b>	O.S : AIX 5.3.0.0



### 3. CICLO DI VITA DEL SOFTWARE

#### 3.1. MODELLO DI RIFERIMENTO

Il modello di riferimento per i processi applicativi realizzati sullo strumento è un ciclo di vita caratterizzato dalla lavorazione degli oggetti software in modalità lineare, dove può essere realizzata anche una concorrenza.



Gli stati che compongono il ciclo di vita base sono i seguenti:

1. "SVILUPPO"
2. "COLLAUDO"
3. "PRODUZIONE"
4. "CESTINO"
5. "ARCHIVIO - SNAPSHOT"

Gli item di configurazione software transitano tra i vari stati all'interno di contenitori, detti pacchetti o packages.

Questo ciclo di vita è utilizzato per lo sviluppo e/o modifica del software in concorrenza.

Sviluppare software in concorrenza significa che un elemento può essere presente, contemporaneamente, in più stati evolutivi nati da esigenze diverse, quale ad esempio quella di manutenzione correttiva di oggetti già inclusi in un'attività di manutenzione evolutiva.

Il check-out di un elemento in una situazione in cui è già presente una sua versione creata da un diverso intervento comporta la generazione di un nuovo ramo contenente versioni intermedie (branching), le quali, per essere integrate con il precedente ramo in un'unica versione finale dell'elemento in questione, dovranno essere poi sottoposte ad un processo - detto d'integrazione (merging) - che si realizza in modo manuale o automatico.

Il merging automatico, consentito solo se le versioni intermedie non hanno subito modifiche sulle stesse porzioni di codice, crea una nuova versione dell'elemento.

Qualora risultino modificate le stesse righe di codice è necessario attivare il merging manuale

Classificazione Consip Public

Gara per la fornitura di prodotti Computer Associates e relativi servizi per il Sistema Informativo Sogei - ID 1543

Appendice 1 al Capitolato Tecnico - Ambiente SCM



(Interactive Merge).

La concorrenza è ammessa solo nell'ambito dello stato logico Sviluppo, e quindi le attività di merging e di consolidamento della versione degli item devono essere svolte prima della promozione nello stato di Collaudo. Il consolidamento di item di configurazione presenti in packages MAC deve essere effettuato prima di consentire il consolidamento di corrispondenti item di configurazione presenti in packages MEV.

Lo stato logico "Cestino" ha l'utilità di contenere i pacchetti con le modifiche che non arriveranno mai in "Produzione".

Lo stato logico "Archivio - Snapshot" ha l'utilità di contenere tutti i package che vogliono essere archiviati relativamente alla "Produzione" oltre ad offrire la possibilità di eseguire degli Snapshot delle viste logiche presenti nel ciclo di vita.

CA Software Change Manager gestisce il software in un ciclo logico, separato dalle reali macchine su cui le applicazioni saranno eseguite.

Al fine di garantire la delivery degli oggetti sottoposti al ciclo di vita, il tool è stato personalizzato ed integrato con l'esecuzione di programmi esterni allo strumento, per garantire la sincronizzazione all'interno di alcune directory remote sul server Unix remoto durante i processi di promote/demote degli oggetti software nei vari stati logici.

Tali directory fungono poi da base di partenza per la realizzazione delle sincronizzazioni sugli ambienti applicativi reali.

### 3.2. TERMINOLOGIA

CA Software Change Manager utilizza una propria terminologia di lavoro che è la seguente:

***Baseline/Insieme Degli Item Di Un Progetto***

Si può identificare nel contenuto iniziale di un progetto applicativo nel repository, conseguente al caricamento tramite processo di *Load*. Le versioni degli item inclusi in tale caricamento sono caratterizzate dalla etichetta "Base" per il relativo package di appartenenza.

***Branch/Versione Ramificata***

Nuova versione creata durante la lavorazione in modalità concorrente, nel caso che sussista già un parallelo prelievo dell'oggetto per altre attività di modifica.

***Form/Maschera***

Videata elettronica che consente la memorizzazione di informazioni associate ad un package.

***Item/Oggetto Del Repository***

Oggetto software presente sul repository.

***Package Group/ Gruppo di Pacchetti***

Aggregazione di un insieme di *package*, i quali possono essere così sottoposti ad alcuni trattamenti mediante un'unica operazione comune (ad es. la promozione).

***Package/Pacchetto***

Contenitore logico che raccoglie gli elementi che veicoleranno la generazione o la variazione del software. Un pacchetto può trovarsi solamente in uno stato alla volta.

***Process/Processo***

Azione applicabile in uno stato.

***Project/Progetto***

Ambiente di lavoro definito in CA Software Change Manager per la gestione della configurazione, corrispondente ad un ambito applicativo completo e autonomo.



<b><i>Repository/Archivio</i></b>	Insieme di directory (virtuali) nelle quali è memorizzato il software di un'applicazione e le relative modifiche.
<b><i>State/Stato</i></b>	Stato evolutivo in cui si trova il software in un project.
<b><i>Tag/Etichetta</i></b>	<p>È un'etichetta che indica lo stato dell'item. Può assumere i seguenti valori:</p> <ul style="list-style-type: none"><li>▪ R = riservata</li><li>▪ N = normal</li><li>▪ M = da fondere (merging)</li><li>▪ D = rimosso logicamente</li></ul>
<b><i>Trunk/Versione In Formato Intero</i></b>	Nuova versione creata durante la lavorazione in modalità <b>lineare</b> , quindi per una modifica successiva di un oggetto precedentemente inserito nel sistema.
<b><i>UDP/Processo Personalizzato</i></b>	Processo CA Software Change Manager per integrare ed eseguire programmi esterni.
<b><i>User Group/Gruppo Utenza</i></b>	Insieme di utenti omogenei cui è consentito di utilizzare gli stessi processi.
<b><i>User/Utenza</i></b>	Utente creato sul sistema CA Software Change Manager che può operare sul progetto di competenza utilizzando i processi definiti dall'amministratore e assegnati al gruppo di appartenenza dell'utente stesso.



### **Version/Versione**

Oggetto differenziato rispetto ad altri di una serie (di *versioni*) cui appartiene e che deriva da un prototipo iniziale (oggetto creato ex novo).

Nel repository CA Software Change Manager ciascun Item è composto di più versioni. Tutte le versioni dell'Item sono visibili tramite gli opportuni processi applicabili.

Le versioni *Base* sono le versioni caricate inizialmente con la *Load* del repository e sono visibili da tutte le viste. Le versioni successive create dopo il caricamento iniziale sono visibili solo nelle viste in cui sono transitate.

Quando un utente registra una versione nuova di un Item, questa diviene visibile solamente fra gli stati che condividono la stessa vista a meno di controlli opportuni effettuati.

Vi sono tre momenti in cui viene creata una nuova versione:

- Durante il primo check-in dell'oggetto, in cui viene creata la versione iniziale "0", equivalente ad una versione Base da un punto di vista logico, infatti, è la prima della serie di versioni possibili per l'oggetto in questione.
- Al momento del check-out Update, quando viene **riservata** una nuova versione per l'oggetto prelevato che viene resa attiva e valida al momento del check-in di un oggetto effettivamente modificato (se l'oggetto non è stato variato la riserva è annullata e la nuova versione è resa disponibile per future modifiche).
- Al momento della Delete, quando viene creata una nuova versione per l'oggetto, caratterizzata dal Tag D, per indicare che l'oggetto in questione non è più utilizzato.

### **View/Vista:**

Vista logica di una porzione del repository, applicabile ad uno stato o più stati.

## **3.3. PROCESSI OPERATIVI**

La gestione del ciclo di vita del software in CA Software Change Manager avviene attraverso dei processi operativi che possono essere di due tipi, Nativi cioè standard del prodotto, oppure personalizzati attraverso gli **UDP (User Defined Process)**.

Di seguito vengono elencati i processi Nativi con una breve descrizione:

<b>Approve</b>	Esegue l'approvazione o il reject di un package.
<b>Checkin</b>	Esegue l'archiviazione sul repository di progetto sia di item nuovi sia esistenti (modificati).
<b>CheckOut</b>	Esegue l'estrazione, e/o la reserve, degli item archiviati. L'estrazione può essere eseguita in <b>Update</b> (l'Item estratto può essere modificato, e in CA Software Change Manager viene riservato) o <b>Browse</b> (l'Item estratto non può essere modificato, e non viene eseguita nessuna operazione).
<b>CompareView</b>	Confronta l'elenco delle versioni presenti in due differenti viste del progetto, evidenziandone corrispondenze e/o





differenze (secondo le opzioni scelte). Consente inoltre, su selezione specifica di un oggetto presente nell'elenco suddetto (purché in formato testo), di confrontare il contenuto delle sue diverse versioni presenti nelle due viste (tramite apertura di finestre di testo giustapposte orizzontalmente).

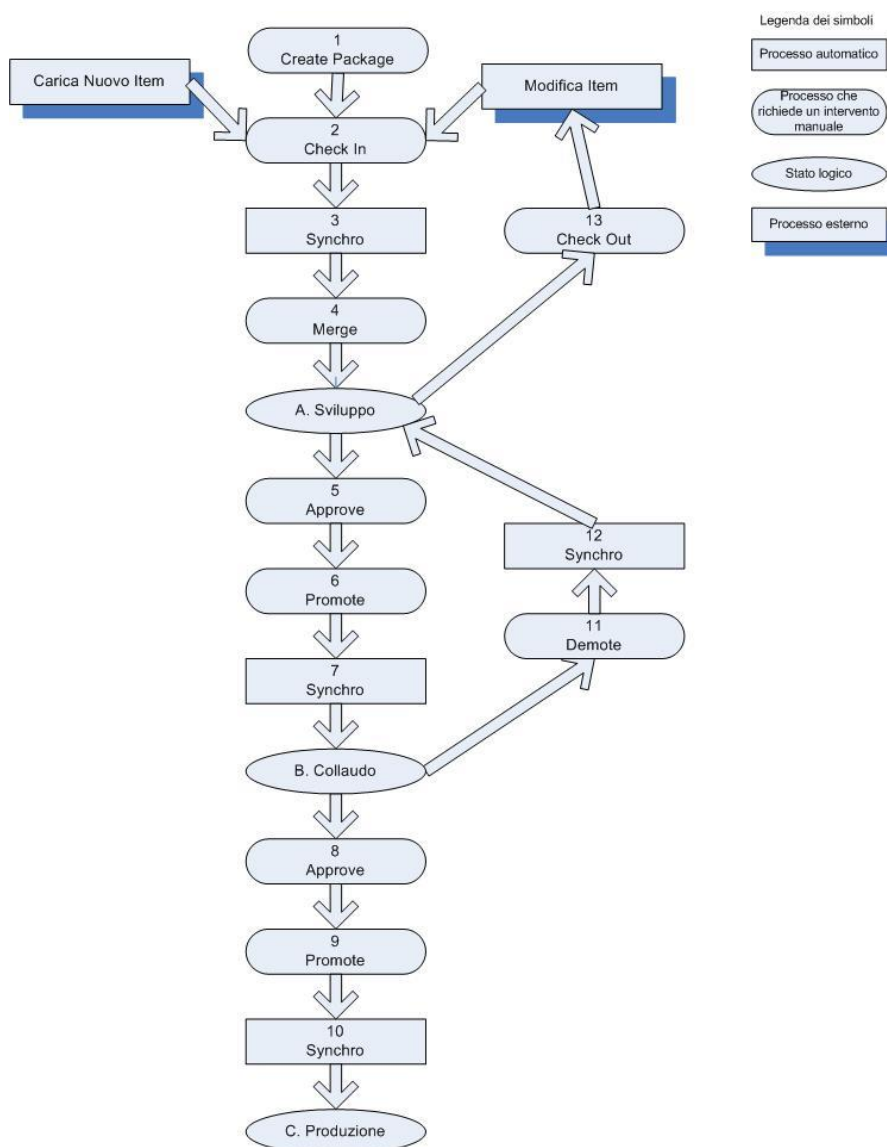
<b>ConcurrentMerge</b>	Crea un'unica versione di un item, in modalità automatica, da due versioni <i>branch</i> dello stesso item. Se le versioni contengono stesse righe di codice modificate, viene creata una versione intermedia, la quale deve essere risolta con il processo " <i>interactive merge</i> ".
<b>Create Package</b>	Crea un package al quale possono essere eventualmente associate le <i>form</i> per l'inserimento d'informazioni sul package stesso.
<b>DeleteVersion</b>	Cancella l'ultima versione di un item (o anche una <i>tag</i> ).
<b>Demote</b>	Esegue la regressione di un package verso uno stato precedente del ciclo di vita.
<b>InteractiveMerge</b>	Crea una versione di un item tramite il merge interattivo, ovvero controllato dall'operatore, di due versioni.
<b>ListVersion</b>	Elenca gli item di una baseline, con annesse alcune informazioni (es: versione, data di modifica, utente, etc).
<b>Notify:</b>	Esegue la notifica automatica, via mail, ad un gruppo di lavoro o a singoli utenti.
<b>Promote:</b>	Esegue la promozione di un package verso lo stato successivo del ciclo di vita.
<b>RenameItem</b>	Rinomina un item già presente in configurazione.
<b>RemoveItem</b>	Cancella logicamente uno o più item da una specifica vista.

Quando si desidera una personalizzazione, come ad esempio si ha necessità di richiamare un programma esterno, eseguire una query sul db, creare un batch con i comandi di CA Software Change Manager, etc, si utilizza allora il processo denominato **UDP (User Defined Process)** dove si inserisce il nome del programma da eseguire con gli eventuali parametri di ingresso, rappresentati dalle variabili di sistema.



### 3.4. FLUSSO ELABORATIVO

Nell'ambito dei progetti di sviluppo software del Ministero Economia e Finanze, posti sotto controllo di configurazione su CA Software Change Manager, il processo di gestione degli item applicativi segue il flusso schematizzato in figura



ed è regolato dall'utilizzo dei seguenti processi:

1. Create Package MAC/Create Package MEV
2. Checkout Item in Concurrent Mode
3. Modifica oggetto, compilazione e test sull'ambiente di sviluppo.
4. Checkin New or Existing Items
5. Concurrent Merge
6. Promote package to collaudo
7. Creazione eseguibili di collaudo
8. Collaudo dell'applicazione

Classificazione Consip Public

Gara per la fornitura di prodotti Computer Associates e relativi servizi per il Sistema Informativo Sogei - ID 1543

Appendice 1 al Capitolato Tecnico - Ambiente SCM



9. Approve
10. Promote package in produzione
11. Creazione eseguibili di produzione
12. Trasferimento in produzione

#### 4. GESTIONE DEL REPOSITORY

Il repository, nell'ambito dell'architettura di prodotto, rappresenta il punto di riferimento per l'acquisizione e l'estrazione degli items applicativi e dei pacchetti utilizzati per veicolare le modifiche software all'interno dei vari stati che compongono il ciclo di vita del software. Si compone di un insieme di directory (virtuali) nelle quali è memorizzato il software di un'applicazione e le relative modifiche.

##### 4.1. TIPI DI FILE

L'ambiente CA Software Change Manager è stato personalizzato in modo da gestire, all'interno di ciascun progetto applicativo, solo i file con un'estensione conosciuta (.sh, .pco, .java, .pdf, etc.). I file con estensione non conosciuta (.pippo, .prova) vengono bloccati al momento del check-in.

Per ogni estensione di file è stata definita la modalità con cui il file stesso deve essere gestito, cioè TESTO o BINARIO. All'atto della promote su collaudo, il trasferimento dell'oggetto sulla macchina Unix di collaudo avviene in modalità ASCII oppure Binario a seconda che il file sia stato definito di tipo Testo o Binario rispettivamente.

#### 5. PERSONALIZZAZIONI

Allo stato attuale sull'ambiente CA Software Change Manager è prevista la gestione delle applicazioni :

- Stipendi
- Anagrafica Unica

Entrambi i progetti applicativi seguono il ciclo di vita del software così come definito nel Modello di Riferimento, ed ognuno dei progetti utilizza sia i processi nativi del prodotto, sia delle integrazioni con dei programmi esterni, detti User Defined Process o UDP, realizzati in Perl, come dettagliato nei seguenti paragrafi, relativi alle attività nei vari stati logici.

##### 5.1. STATO LOGICO SVILUPPO

Lo stato logico di Sviluppo è il punto di ingresso per le differenti modifiche (evolutive e correttive) che si vogliono apportare agli item del software. All'interno di questo stato viene gestita inoltre la concorrenza tra più versionamenti dello stesso item. Tale concorrenza dovrà essere risolta prima della promozione nello stato logico di Collaudo.

All'interno di questo stato vengono svolte le seguenti attività :

- **Create Package MAC** : Creazione di un pacchetto MAC, utilizzato per veicolare nei vari stati logici gli items software legati ad una modifica correttiva. Questa attività è realizzata dall'UDP CreaPack.pl
- **Create Package MEV** : Creazione di un pacchetto MEV, utilizzato per veicolare nei vari stati logici gli items software legati ad una modifica evolutiva. Questa attività è realizzata dall'UDP CreaPack.pl
- **Checkout Items in Concurrent Mode** : Estrazione di uno o più item verso una area di lavoro locale utilizzata per eseguire le modifiche richieste. L'estrazione degli item avviene in modalità concorrente per un dato package MAC/MEV. All'interno di tale processo viene eseguita una verifica ed una eventuale segnalazione degli item che sono stati presi in carico da altri package che siano attualmente in lavorazione in



Sviluppo e Collaudo. Questa attività è integrata con i seguenti script Perl : **warningCheckout.pl** e **controlCheckout.pl**

- **Checkin New or Existing Items** : Inserimento di uno o più items da una area di lavoro locale per aggiornamento del software. Possono essere inseriti nuovi item o item precedentemente estratti per la modifica. Gli item che non presentano differenze in fase di inserimento vengono eliminati (con il loro versionamento) dal pacchetto. Viene eseguito un controllo sulla validità delle estensioni files consentite nel progetto. Questa attività è integrata con i seguenti scripts Perl : **checkExtension.pl** e **controlCheckin.pl**
- **Concurrent Merge** : Esegue il Concurrent Merge, quindi la creazione di una versione temporanea per ogni item presente in un pacchetto. All'interno di tale processo viene eseguito un controllo che impone la priorità di una MAC su una MEV. Questa attività è realizzata dall'UDP **controlConcurrentMerge.pl**
- **Interactive Merge Version** : Esegue l'Interactive Merge, quindi il consolidamento di una versione per ogni item presente in un pacchetto. All'interno di tale processo viene eseguito un controllo che impone che un pacchetto MEV non può eseguire tale processo se la MAC non è stata approvata/collaudata e passata in Produzione. Questa attività è integrata con i seguenti scripts Perl : **controlInteractiveMerge.pl**
- **Promote Package to Collaudo** : Esegue la promozione del package in nello Stato di Collaudo. All'interno di tale processo viene eseguito un controllo che impone la non concorrenza tra le MAC e tra le MEV e il rispetto del versionamento degli item con quelli presenti in Produzione. Viene inoltre effettuata la sincronizzazione delle directory remote di collaudo. Questa attività è integrata con i seguenti scripts Perl : **controlPromoteCollaudo.pl** e **NewSincroRemote.pl**

## 5.2. STATO LOGICO COLLAUDO

Lo stato logico di Collaudo è l'ambiente in cui vengono consolidate le differenti modifiche (evolutive e correttive) che vengono apportare agli item del software prima del passaggio definitivo nello stato di Produzione.

All'interno di questo stato vengono svolte le seguenti attività :

- **Approve/Reject Package** : Approvazione o rifiuto di un package selezionato. Tale processo di approvazione è utilizzabile dai responsabili del Collaudo e della Produzione che dovranno verificare la correttezza delle modifiche apportate agli item del package in questione ed approvare/rifiutare la messa in produzione delle modifiche. Non è integrato con alcun UDP personalizzato.
- **Demote package to Sviluppo** : Esegue la regressione nello stato di Sviluppo di un package di cui sia stata rifiutata la promote nello stato finale di Produzione. Risincronizza inoltre le directory remote di collaudo. Questa attività è integrata con i seguenti scripts Perl : **controlDemoteSviluppo.pl**, **deleteltemColl.pl** e **regressioneSvilPackage.pl**
- **Promote Package to Produzione** : Esegue la promozione del package in nello Stato di Produzione. Tale promozione è vincolata dal buon esito dei due processi di approvazione (approvazione a doppia chiave) da parte del Responsabile di Collaudo e del Responsabile di Produzione. Viene inoltre effettuata la sincronizzazione delle directory remote di produzione. Questa attività è integrata con i seguenti scripts Perl : **controlPromote.pl** e **newSincroRemote.pl**

## 5.3. STATO LOGICO PRODUZIONE

Lo stato logico di Produzione è l'ambiente finale nell'ambito del ciclo di vita del software, cui approdano definitivamente le differenti modifiche (evolutive e correttive) che vengono

Classificazione Consip Public

Gara per la fornitura di prodotti Computer Associates e relativi servizi per il Sistema Informativo Sogei - ID 1543

Appendice 1 al Capitolato Tecnico - Ambiente SCM



apportare agli item del software.

All'interno di questo stato vengono svolte le seguenti attività :

- **Create Snapshot** : Tale processo serve alla creazione di Snapshot (viste logiche) il cui obiettivo è quello di 'fotografare' la release software in Produzione e che possono essere solo consultate nello stato logico di Archiviazione - Snapshot. Questa attività è realizzata attraverso i processi nativi del prodotto.
- **Create Report - Baseline Latest Version** : Viene creato un report in output al programma e in un file fisico che viene segnalato dove si elencano tutte le ultime versioni degli item presenti nello stato di Produzione. Questa attività è realizzata dall'UDP *prodBaseline.pl*
- **Checkout Baseline Latest Version + Create Report** : Viene estratta tutta la baseline presente nello stato logico di Produzione con le ultime versioni degli item in un path predefinito che viene segnalato e Viene creato un report in output al programma e in un file fisico che viene segnalato dove si elencano tutte le ultime versioni degli item presenti nello stato di Produzione. Questa attività è realizzata dall'UDP *checkoutLatestVersion.pl* ed è integrata con i seguenti scripts Perl : *ProdBaseline.pl*
- **Demote Package to Collaudo** : E' una attività che viene utilizzata solo eccezionalmente e che può essere eseguita dal solo amministratore di CA Software Change Manager. Viene regredito il package nello stato di Collaudo e contestualmente viene sincronizzata la versione precedente degli item, sulle directory remote di produzione. Questa attività è realizzata dall'UDP *CheckItemColl.pl* ed è integrata con i seguenti scripts Perl : *controllLastVerProd.pl*

## 6. SINCRONIZZAZIONE AMBIENTI UNIX REMOTI

La soluzione CA Software Change Manager è stata personalizzata con l'ausilio di scripts Perl in particolare per realizzare la sincronizzazione di directory remote durante la fasi di Promote e Demote negli ambienti di Collaudo e Produzione, in modo da rendere automatica la sincronizzazione degli items software tra il Repository e la macchina Unix di collaudo in concomitanza alle operazioni movimentazione degli items software.

La macchina Unix di collaudo, rispecchiando la configurazione software applicativa, controllata con lo strumento CA Software Change Manager viene così utilizzata come punto di riferimento per il deploy applicativo sull'ambiente di produzione reale.

La logica degli scripts di sincronizzazione segue quella di deploy dell'applicazione specifica ed è guidata dall'utilizzo di file di configurazione specifici per ogni ambito applicativo. I paragrafi che seguono dettagliano le personalizzazioni inserite in funzione dei progetti applicativi posti sotto il controllo di CA Software Change Manager.



## 6.1. SINCRONIZZAZIONE STIPENDI

Promote tra stati logici CA Software Change Manger del ciclo di vita		Scripts perl di Sincronizzazione
DA	A	
SVILUPPO	COLLAUDO	ControlPromote.pl NewSincroRemote.pl NewReassignePermission.pl NewSincroMirata.pl NewChangePermission.pl
COLLAUDO	PRODUZIONE	CheckApprove.pl ControlPromote.pl NewSincroRemote.pl NewDelete_Item.pl

Demote tra stati logici CA Software Change Manger del ciclo di vita		Scripts perl di Sincronizzazione
DA	A	
PRODUZIONE	COLLAUDO	CheckItemColl.pl
COLLAUDO	SVILUPPO	controlDemoteSviluppo.pl NewDelete_Item.pl NewCopyMirata.pl

## 6.2. SINCRONIZZAZIONE ANAGRAFICA UNICA

Promote tra stati logici CA Software Change Manger del ciclo di vita		Scripts perl di Sincronizzazione
DA	A	
SVILUPPO	COLLAUDO	ControlPromote.pl NewSincroRemote.pl
COLLAUDO	PRODUZIONE	ControlPromote.pl NewSincroRemote.pl

Demote tra stati logici CA Software Change Manger del ciclo di vita		Scripts perl di Sincronizzazione
DA	A	
PRODUZIONE	COLLAUDO	ControlLastVerProd.pl CheckItemColl.pl
COLLAUDO	SVILUPPO	controlDemoteSviluppo.pl Delete_ItemColl.pl RegressionSvilPackage.pl