



consip

Gara europea per l'acquisizione di servizi di sviluppo, manutenzione, gestione applicativa, supporto tematico, change management e formazione per il Sistema Informativo Service Personale Tesoro (SPT)

APPENDICE 6 AL CAPITOLATO TECNICO

**Gestione della configurazione:
strumenti, standard e modalità d'uso**



INDICE

1	INTRODUZIONE	3
2	GLOSSARIO	4
3	ACRONIMI	5
4	IL CICLO DI VITA DEL SOFTWARE	6
4.1	MODELLO DI "ANALISI RICHIESTE"	7
4.1.1	STATO "RICHIESTO"	8
4.1.2	STATO "IN ANALISI"	9
4.1.3	STATO "ANALIZZATO"	9
4.1.4	STATO "SCHEDULATO"	9
4.1.5	STATO "CANCELLATO"	9
4.1.6	STATO "SOSPESO"	10
4.2	MODELLO DI "SVILUPPO E MODIFICA DEL SW IN CONCORRENZA"	11
4.2.1	STATO "SVILUPPO/MEV"	13
4.2.2	STATO "MANUTENZIONE [MAC]"	13
4.2.3	STATO "COLLAUDO"	14
4.2.4	STATO "PRODUZIONE"	15
5	RUOLI E RESPONSABILITÀ	16
5.1	RESPONSABILE DI PROGETTO	16
5.2	AMMINISTRATORE DEL CCM	16
5.3	RESPONSABILE DELLO SVILUPPO	17
5.4	RESPONSABILE DELLA MANUTENZIONE	17
5.5	SVILUPPATORE/MANUTENTORE	17
5.6	RESPONSABILE DI COLLAUDO	18
5.7	RESPONSABILE DI PRODUZIONE	18
5.8	GESTORE DEL PRODOTTO DI CCM	18



1 INTRODUZIONE

Scopo del presente documento è definire le linee guida per facilitare l' utilizzo del prodotto attualmente impiegato sul Sistema SPT per gestire la configurazione.

L'Amministrazione/Consip si riserva di poter cambiare tale strumento durante il periodo di durata contrattuale e, in questo caso, il Fornitore deve garantire il corretto trasferimento dei dati nel nuovo strumento.

Il prodotto di Configuration & Change Management (**CCM**) per il progetto Service Personale Tesoro (SPT) utilizzato è Harvest della Computer Associates.



2 GLOSSARIO

Ambiente	Vedi Environment.
Approve	Processo di approvazione. Autorizzazione di uno o più utenti che permette ad un package di essere successivamente spostato.
Checkin archivio.	Inserimento di un elemento software modificato nel relativo
Checkout	Estrazione di un elemento software, per modificarlo od ispezionarlo.
Demote	Regressione di un package da uno stato ad un altro precedente.
Environment progetto).	Ambiente dove viene gestito il software (è assimilabile ad un
Form	Videata elettronica che gestisce le informazioni associate ad un package.
Manutentore	Persona deputata a modificare il software.
Move Package	Spostamento di uno o più package verso un altro ambiente.
Package	Contenitore logico che contiene gli elementi che veicoleranno la generazione o variazione del software
Package Group	Aggregazione di un gruppo di packages, i quali possono essere sottoposti a particolari operazioni.
Process	Azione applicabile ad uno stato.
Promote	Promozione di un package da uno stato ad un altro.
Repository	Insieme di directory, dove viene registrata una versione software, di un'applicazione con le rispettive modifiche (delta).
State	Stato evolutivo del software contenuto in un ambiente.
Sviluppatore	Persona deputata a sviluppare il software.
User	Utente creato sul sistema Harvest.
User Group	Utente assegnato ad una gerarchia di utenti in Harvest.
View	Vista logica di una porzione del repository, applicabile ad uno stato.



3 ACRONIMI

CCM	Configuration & Change Management
CdV	Ciclo di Vita
CDEV	Concurrent Development (sviluppo in concorrenza)
CI	Configuration Item
HW	Hardware
N.A.	Non Applicabile
SW	Software
TNR	Test di Non Regressione del Software
MAC	(Manutenzione) Migliorativa, Adeguativa, Correttiva
MEV	Modifica Evolutiva.

4 IL CICLO DI VITA DEL SOFTWARE

Nel processo di definizione del Ciclo di Vita per lo sviluppo e la manutenzione del software di SPT, è necessario considerare alcune attività fondamentali, quali ad esempio:

1. L'identificazione dell'ambiente o environment:
rappresenta l'ambiente, cioè l'apparato logico in cui viene gestita un'applicazione software. Per ogni nuovo progetto si prevede la definizione di un apposito environment.
2. La definizione degli stati evolutivi:
all'interno d'ogni environment, si definiscono gli stati evolutivi del progetto software in esame, in altre parole le condizioni che esso può assumere durante il suo ciclo di vita.
3. Definizione dei ruoli e delle responsabilità:
 - Profili di sicurezza, determinanti il livello d'autorizzazione prestabilito per poter attuare operazioni sugli oggetti gestiti.
 - Procedure per la gestione delle richieste di modifiche al software esistente o sviluppo di nuove applicazioni, determinanti l'assegnazione di specifiche responsabilità per ogni singolo passo operativo.
4. Il controllo delle modifiche apportate agli elementi configurati (oppure ai nuovi elementi) in termini di:
 - Chi ha eseguito la modifica/inserimento di un elemento.
 - Cosa è stato modificato.
 - Quando è stato modificato/inserito.
5. La gestione della documentazione:
Per documentazione s'intende tutti i documenti prodotti e utilizzati nelle varie fasi progettuali e manutentive come, ad esempio, l'analisi dei requisiti, documenti architetture, analisi funzionale, segnalazioni d'errore, interventi di manutenzione, etc. Tali oggetti possono essere visti anch'essi come item di configurazione da gestire.

Per SPT sono stati identificati due cicli di vita:

Analisi delle Richieste.

Richiesta di Modifica al software esistente (MAC).

Richiesta di nuove funzionalità (Sviluppo) o di Manutenzione evolutiva (MEV).

Ciclo di vita dello sviluppo e modifica del software, gestito in concorrenza. In particolare lo sviluppo in concorrenza consente la gestione di un item presente in più stati evolutivi, con i seguenti vincoli:

- non si possono avere in collaudo gli stessi oggetti anche se di versioni diverse;
- il check-out degli oggetti MEV deve essere fatto sempre in concorrenza;
- le MAC devono avere priorità.

4.1 MODELLO DI “ANALISI RICHIESTE”

Questo ciclo di vita potrà essere utilizzato per gestire, monitorare e analizzare le richieste di:

- Sviluppo software e manutenzione evolutiva.
- Manutenzione al software esistente, intese come modifiche:
 1. Migliorative.
 2. Adeguative.
 3. Correttive.

Per il ciclo di vita qui esposto sono stati identificati i seguenti stati evolutivi:

- “RICHIESTO”
- “IN ANALISI”
- “ANALIZZATO”
- “SCHEDULATO”
- “CANCELLATO”
- “SOSPESO”

Di seguito illustriamo, graficamente, il ciclo di vita in questione al quale segue una descrizione dettagliata di ciascuno stato.

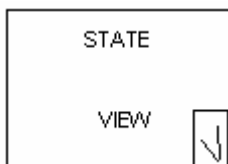
Legenda



Promote (Promozione della richiesta da uno stato ad un altro).



Demote (Regression di una richiesta verso uno stato, per rianalizzarlo)

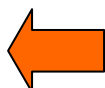


Il rettangolo rappresenta uno dei possibili stati, dove

- STATE, indica il nome dello stato
- VIEW, indica il nome della view, cioè la vista logica sul repository degli oggetti.
- Il rettangolino in basso a destra indica che in quello stato è prevista l'approvazione per poter passare ad uno stato successivo.



Uscita dal Ciclo di vita “Analisi Richieste” (verso il ciclo “Sviluppo e Modifica del Software In Concorrenza”).



Fase iniziale del Ciclo di vita “Analisi Richieste”, determinato da una richiesta sviluppo/modifica.

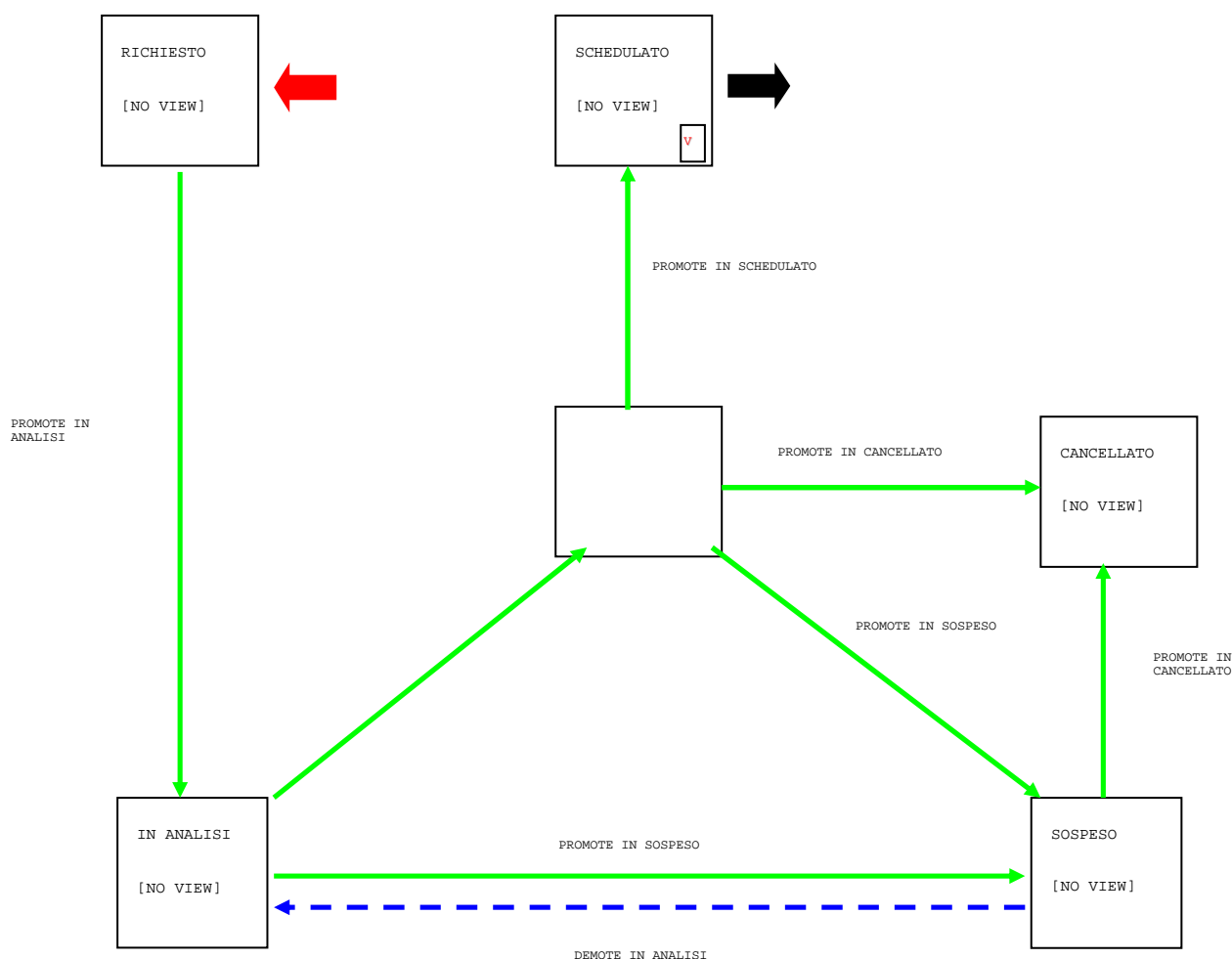


Fig. 1 - Analisi Richieste

4.1.1 STATO “RICHIESTO”

In questa fase il Responsabile dello Sviluppo o della Manutenzione può effettuare una creazione di un package, relativo ad una richiesta di:

1. Manutenzione al software esistente: migliorativa, adeguativa, correttiva (MAC).
2. Sviluppi di nuove funzionalità e/o manutenzione evolutiva (MEV).

Il package creato in questo stato, seguirà a seconda che si tratta di una MAC o di un nuovo sviluppo/MEV la seguente nomenclatura:

1. RICHIESTA MANUTENZIONE:
 - MAN#AREA#FUNZ-AA-xxxx, dove:
 - MAN, identifica la tipologia d'intervento (richiesta di manutenzione).
 - AREA, identifica l'area gestionale.



- FUNZ, identifica l'obiettivo/progetto/classe d'intervento.
- AA, rappresenta l'anno corrente.
- xxxx, rappresenta il progressivo della richiesta (ogni inizio anno viene riportato a zero).

2. RICHIESTA NUOVO SVILUPPO/MEV:

- SVI#AREA#AREA#FUNZ-AA-xxxx, dove:
 - SVI, identifica la tipologia d'intervento (richiesta di sviluppo/mev).
 - AREA, identifica l'area gestionale.
 - FUNZ, identifica l'obiettivo/progetto/classe d'intervento.
 - AA, rappresenta l'anno corrente.
 - xxxx, rappresenta il progressivo della richiesta (ogni inizio anno viene riportato a zero).

Il package viene successivamente spostato, con un'operazione di promote, nello stato "IN ANALISI".

4.1.2 STATO "IN ANALISI"

In questo stato, si effettua l'analisi di tutte le problematiche che la richiesta comporta. La richiesta può essere

- spostata nello stato di "SOSPESO"
- spostata nello stato di "ANALIZZATO".

4.1.3 STATO "ANALIZZATO"

Questo stato rappresenta un'area di deposito temporaneo, prevista per tutte le richieste, la cui attività d'analisi risulta completata e che sono in attesa di un successivo passaggio allo stato di

- "SCHEDULATO"
- "CANCELLATO"
- "SOSPESO".

4.1.4 STATO "SCHEDULATO"

Questo stato è utilizzato per spostare la richiesta di sviluppo/modifica nell'ambiente operativo di manutenzione o sviluppo/MEV. Operativamente, ciò si effettua tramite una "move package".

Si dà origine in questo modo al ciclo di vita "Sviluppo e Modifica del Software In Concorrenza".

Lo spostamento può avvenire solo dopo approvazione del Responsabile dello sviluppo o della MAC.

4.1.5 STATO "CANCELLATO"

Questo stato contiene tutte le richieste che sono state ritenute non valide per il processo di sviluppo o di manutenzione e quindi eliminate definitivamente dal ciclo di vita.



4.1.6 STATO “SOSPESO”

Questo stato contiene tutte le richieste, ritenute momentaneamente non realizzabili. Esse saranno oggetto di un successivo spostamento nello stato

- “CANCELLATO”, qualora si ritengono definitivamente non realizzabili
- “IN ANALISI”, qualora si ritenga di dover effettuare ulteriori approfondimenti.

4.2 MODELLO DI “SVILUPPO E MODIFICA DEL SW IN CONCORRENZA”

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

Sviluppare software in concorrenza, significa che un item 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 checkin di un elemento in una situazione in cui è già presente una sua versione creata da un diverso intervento comporta la generazione di versioni intermedie (branching), le quali, per essere integrate 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.

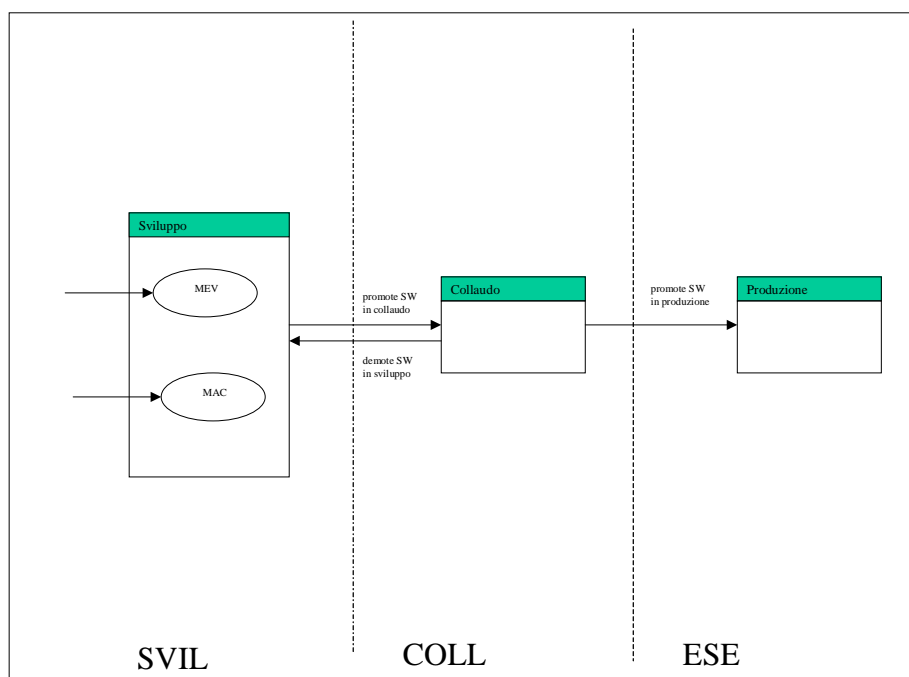
Il merging automatico, consentito solo se le versioni intermedie non hanno subito modifiche sulle stesse porzioni di codice, crea una nuova versione dell'elemento.

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

Sono stati identificati i seguenti stati evolutivi del codice sorgente:

- “SVILUPPO/TEST (MEV)”
- “MANUTENZIONE/TEST (MAC)”
- “COLLAUDO”
- “PRODUZIONE”

Di seguito illustriamo, graficamente, il ciclo di vita in questione al quale segue una descrizione di ciascuno stato.




 consip	<i>Gara europea per l'acquisizione di servizi di sviluppo, manutenzione, gestione applicativa, supporto tematico, change management e formazione per il Sistema Informativo Service Personale Tesoro (SPT)</i>
---	--

Fig. 2 - Ciclo di Vita dello Sviluppo e/o Modifica Del Software



4.2.1 STATO “SVILUPPO/MEV”

Al momento dell’inserimento di un package nello stato di sviluppo/MEV il Responsabile dello “Sviluppo” riceve una mail indicante alcune informazioni quali:

- Nome dell’applicazione e/o progetto
- Tipologia di intervento:
 - Manutenzione evolutiva
 - Sviluppo
- Altre informazioni.

Successivamente egli notifica tramite mail agli sviluppatori interessati l’assegnazione dell’attività di modifica/sviluppo.

Lo sviluppatore esegue:

- un checkout, in modalità concorrente, di uno o più item verso una propria area di lavoro per eseguire lo sviluppo o le modifiche.

Qualora un altro sviluppatore richieda il checkout degli stessi item, riceve una notifica dell’esistenza di un’attività concorrente su quegli stessi oggetti.

- il “test unitario” degli oggetti modificati e /o creati.
Se tale test ha esito positivo esegue il
 - “checkin” del modulo
 - notifica, tramite mail, al Responsabile di Sviluppo che il sorgente è pronto per essere promosso nello stato di “COLLAUDO”.

Il Responsabile dello Sviluppo, una volta ricevuta notifica della conclusione di tutte le attività di sviluppo, esegue l’approvazione del package che deve essere promosso.

4.2.2 STATO “MANUTENZIONE [MAC]”

Il Responsabile della “MANUTENZIONE”, riceve una mail indicante alcune informazioni quali:

- Nome dell’applicazione e/o progetto
- Tipologia di intervento
- Altre informazioni.

Successivamente invia una mail agli sviluppatori interessati, con la quale assegna ufficialmente la richiesta di manutenzione.

Il gruppo di manutenzione esegue:

- un checkout, in modalità concorrente, di uno o più file verso una propria area di lavoro, per eseguire le modifiche al codice sorgente.
Qualora un altro sviluppatore richieda il checkout degli stessi item, riceve una notifica dell’esistenza di un’attività concorrente su quegli stessi oggetti.
- il “test unitario” degli oggetti modificati e/o creati.
Se tale test ha esito positivo, esegue il:



- “checkin” del modulo
- notifica, tramite mail, al Responsabile della Manutenzione che il sorgente è pronto per essere promosso nello stato di “COLLAUDO”.

Il Responsabile della Manutenzione, una volta ricevuta notifica della conclusione di tutte le attività di manutenzione, esegue l’approvazione del package che deve essere promosso.

Nel caso in cui siamo in presenza di una “manutenzione urgente” ed esistono gli stessi oggetti in collaudo, si esegue:

- Una notify al Responsabile di Sviluppo che occorre promuovere una manutenzione urgente. Se questi ritenesse critico il rallentamento delle proprie attività già avanzate dovrà immediatamente comunicarlo al responsabile di Test Manutenzione.
- Contestualmente, in Collaudo è eseguita una demote del package di sviluppo (contenente gli stessi sorgenti della manutenzione urgente) verso lo stato di sviluppo e la sua versione integrata è cancellata. Prima della cancellazione l’effettuazione di un backup automatico su area d’appoggio consente la successiva reintegrazione nello stato di sviluppo, onde permettere l’integrazione delle variazioni dei sorgenti coinvolti nella modifica urgente.
- Il checkout del package, in modalità ‘read-only’, verso una propria area di lavoro per poter generare i corrispondenti eseguibili.
- Il test globale sul software eseguibile generato.
- Se tale test ha esito positivo, il Responsabile del Test Manutenzione esegue
 - l’approvazione del software che deve essere promosso
 - la notifica, tramite una mail, al Responsabile di Manutenzione che il software è pronto per essere integrato (merge automatico o manuale).
- Se tale test ha esito negativo, il Responsabile del Test Manutenzione esegue
 - la notifica, con una mail, al Responsabile di Manutenzione, che il software sarà riportato in tale stato per essere nuovamente modificato
 - la regressione del codice sorgente verso lo stato di Manutenzione.

4.2.3 STATO “COLLAUDO”

Al Responsabile del “COLLAUDO” sarà inviata una mail indicante le informazioni sul software da promuovere in tale stato.

Il Responsabile di Collaudo esegue:

- La promozione del software da sottoporre a testing, con i seguenti vincoli:
 - 1) se in collaudo c’è un oggetto proveniente da MAC,
 - non deve essere permesso il check-out dello stesso oggetto in MAC;
 - non deve essere permessa la promozione di un package contenente lo stesso oggetto e proveniente da MEV. La promozione si farà solo quando l’oggetto in COLLAUDO sarà passato in PRODUZIONE.
 - 2) Se in COLLAUDO c’è un oggetto proveniente da MEV,



- deve essere permesso il check-out dello stesso oggetto in MAC, e la promozione in COLLAUDO con conseguente demote del package contenente l'oggetto che stava in SVILUPPO. La promozione del package si farà solo quando l'oggetto proveniente da MAC sarà passato in PRODUZIONE, e sarà obbligatorio il merge tra i due oggetti.
 - deve essere permesso il check-out dello stesso oggetto in MEV, ma deve essere bloccata la promozione in COLLAUDO. La promozione dell'oggetto si farà solo quando lo stesso sarà passato in PRODUZIONE e sarà obbligatorio il merge tra i due oggetti.
- Un checkout, in modalità 'read-only', per poter generare i corrispondenti eseguibili
- I "test funzionali/prestazionali" sul software eseguibile generato
- Se tale test ha esito positivo, il Responsabile di Collaudo esegue
 - l'approvazione del software che deve essere promosso
 - la notifica, con una mail, al Responsabile del "COLLAUDO SUPERATO", che il software è pronto per essere promosso in tale stato.
- Se tale test ha esito negativo, il Responsabile di Collaudo esegue
 - la notifica, con una mail, al Responsabile di Sviluppo o di Manutenzione, che il package sarà riportato in tali stati per essere nuovamente modificato
 - la regressione del package verso lo stato di sviluppo o di manutenzione.

4.2.4 STATO "PRODUZIONE"

Al Responsabile di "PRODUZIONE" verrà inviata una mail indicante le informazioni sul software promosso in tale stato.

Il responsabile esegue:

- La promozione del software proveniente dallo stato "COLLAUDO"
- Un checkout dell'ultima versione, in modalità 'read-only', per poter generare i corrispondenti eseguibili.



5 RUOLI E RESPONSABILITÀ

Di seguito sono riportate le responsabilità, in relazione ai diversi ruoli ricoperti nel processo di CCM. Tali ruoli descrivono la visione funzionale dell'organizzazione del lavoro rispetto al processo descritto. Non hanno una corrispondenza aprioristicamente biunivoca con la realtà delle strutture di risorse operanti ma devono, anzi, essere definite di volta in volta rispetto alle reali esigenze. Nella pratica ciò può anche comportare che una stessa persona o gruppo ricopra più di uno dei ruoli seguenti o, viceversa, che più persone/gruppi adempiano le funzioni d'uno specifico ruolo.

- Responsabile di Progetto
- Amministratore del CCM
- Responsabile di Sviluppo
- Responsabile di Manutenzione
- Responsabile di Test Sviluppo
- Responsabile di Test Manutenzione
- Sviluppatore
- Manutentore
- Responsabile di Collaudo
- Responsabile di Produzione
- Gestore del prodotto CCM.

5.1 RESPONSABILE DI PROGETTO

DOVE OPERA: Nell'ambito del progetto assegnato, con funzioni di coordinamento, seguendo attivamente tutte le fasi operative fino allo stato di esercizio.

COSA FA: Attribuisce i profili professionali alle risorse allocate sul progetto (e i relativi permessi di sicurezza).

Autorizza, di concerto con le altre risorse, la promozione del software da uno stato ad un altro.

Monitorizza lo stato delle anomalie.

Verifica che le consegne degli elementi software, identificate univocamente, rispondano ai seguenti requisiti:

1. **Completezza** (Verifica degli elementi consegnati).
2. **Qualità** (Motivazione e descrizione della modifica).
3. **Affidabilità** (Esecuzione dei test unitari).

5.2 AMMINISTRATORE DEL CCM

DOVE OPERA: Opera in tutti gli ambienti.

PREREQUISITI: Ha una consapevolezza approfondita delle problematiche di CCM e del prodotto di configurazione. Opera in accordo alle direttive in merito agli standards di qualità software. Non gli è necessaria un'elevata conoscenza delle problematiche interne alle applicazioni.



COSA FA: Si occupa della definizione dei profili professionali compatibilmente con i requisiti predefiniti e secondo le richieste del settore applicativo.
Configura gli ambienti di CCM, in altre parole mette sotto controllo di configurazione le nuove aree applicative.
Gestisce le eventuali variazioni dei cicli di vita esistenti.

5.3 RESPONSABILE DELLO SVILUPPO

DOVE OPERA: Nello stato di sviluppo.

PREREQUISITI: Ha una consapevolezza delle problematiche di CCM e conoscenze del prodotto di configurazione. Opera in accordo alle direttive in merito agli standards di qualità software. È ha una padronanza completa delle problematiche interne alle applicazioni software (che sta gestendo).

COSA FA: Crea un package.
Monitorizza lo stato d'avanzamento dello sviluppo.
Autorizza il passaggio alla fase di test dopo aver verificato il completamento delle attività di sviluppo.

5.4 RESPONSABILE DELLA MANUTENZIONE

DOVE OPERA: Nello stato di manutenzione.

PREREQUISITI: Ha una consapevolezza delle problematiche di CCM e conoscenze del prodotto di configurazione. Opera in accordo alle direttive in merito agli standards di qualità software. È ha una padronanza completa delle problematiche interne alle applicazioni software (che sta gestendo).

COSA FA: Crea un package.
Monitorizza lo stato d'avanzamento dell'intervento.
Autorizza il passaggio alla fase di test dopo aver verificato il completamento delle attività di manutenzione.

5.5 SVILUPPATORE/MANUTENTORE

DOVE OPERA: Opera negli ambienti dedicati all'editing (sviluppo e modifica del software rispettivamente) e negli ambienti di test, in base alle direttive del relativo responsabile.

PREREQUISITI: Ha una conoscenza tecnica delle tematiche del progetto e una consapevolezza (anche se non approfondita) delle problematiche di CCM.

COSA FA: Genera o modifica il software.
Esegue la fase di test unitario e/o globale.
Notifica al responsabile dello sviluppo/manutenzione la fine delle sue attività.



5.6 RESPONSABILE DI COLLAUDO

DOVE OPERA: Opera nello stato di collaudo.

COSA FA: Presiede attivamente alle fasi di test del software, eseguite dal gruppo di collaudo al fine di valutare le caratteristiche funzionali e prestazionali dell'applicazione.

Attesta l'esito finale delle operazioni di collaudo.

Notifica l'esito al Responsabile di Produzione, in caso positivo.

Notifica l'esito al Responsabile di Sviluppo o Manutenzione, in caso negativo.

5.7 RESPONSABILE DI PRODUZIONE

DOVE OPERA: Opera nello stato di produzione.

COSA FA: Rende corrente ed esecutivo il software proveniente dall'ambiente di Collaudo.

A fronte d'errori invoca processi di ripristino, in collaborazione con il Gestore del Prodotto di CCM coordina le attività di "delivery del software" in produzione.

5.8 GESTORE DEL PRODOTTO DI CCM

DOVE OPERA: Opera in tutti gli ambienti.

PREREQUISITI: Ha una consapevolezza delle problematiche tecniche del prodotto di configurazione.

COSA FA: Manutiene, reinstalla il prodotto e predispone le basi dati affinché siano realizzati i test durante le fasi di collaudo e la manutenzione delle applicazioni.

Monitorizza l'uso del prodotto di configurazione.

L'attribuzione delle specifiche azioni può essere riassunta nel seguente scenario:

- Responsabile Di Progetto
 - Tutte le autorizzazioni ad effettuare le azioni specifiche per un progetto.
- Amministratore di CCM
 - Tutte le autorizzazioni, per tutti i progetti.
- Responsabile di Sviluppo e Manutenzione
 - Checkout
 - Checkin
 - Approvazione.
- Responsabile di Test Sviluppo e Test Manutenzione
 - Promozione
 - Checkout read-only
 - Approvazione.
- Sviluppatore/Manutentore
 - Checkout



- Checkin.
- Responsabile di Collaudo
 - Promozione
 - Checkout
 - Approvazione.
- Responsabile di Produzione
 - Promozione
 - Checkout read-only.