

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA
Corso di Laurea in Ingegneria e Scienze Informatiche Magistrale

PROGETTAZIONE ED IMPLEMENTAZIONE DI UNA
TRANSAZIONE IN RADIOFREQUENZA PER UN
SISTEMA SAP-EWM

Elaborata nel corso di: Sistemi Informativi

Tesi di Laurea Magistrale di:
MISSERE MATTIA

Relatore:
Chiar.mo Prof. MATTEO
GOLFARELLI

ANNO ACCADEMICO 2017-2018 (SESSIONE II)

PAROLE CHIAVE

SAP

Extended Warehouse Management

Radio Frequency Environment

ABAP

Customizing

Indice

Introduzione	1
1 Le tecnologie	5
1.1 Sistema ERP	5
1.1.1 Vantaggi e svantaggi dei sistemi ERP	6
1.2 SAP	7
1.2.1 Architettura SAP	8
1.2.2 Tipologie di sistemi SAP	9
1.2.3 Transazioni SAP	11
1.3 SAP-EWM	12
1.3.1 Caratteristiche principali EWM	13
1.3.2 Architettura EWM	13
1.3.3 Unità organizzative	14
1.4 Linguaggio di programmazione ABAP	15
1.4.1 Tipologie di programmi	16
1.4.2 Struttura di un programma	16
1.4.3 Elementi del linguaggio	16
1.5 RF Environment	20
1.5.1 Architettura del RF Environment	21
1.5.2 Definizioni fondamentali RF Environment	21
2 Il processo di magazzino: ricezione merce	23
2.1 Soluzione WM per la ricezione merce	23
2.2 Movimenti multi-step di EWM	24
2.3 Comunicazione tra sistemi ERP ed EWM per le consegne	26
2.4 Soluzione EWM per la ricezione merce	27
2.5 Serial Number Profile	28
2.6 Integrazione dei numeri seriali nelle consegne	30
2.7 Inserimento dei numeri seriali a sistema a livello di item della delivery	31
2.7.1 Processo as is	31
2.7.2 Processo to be	33

3	Progetto	35
3.1	Requisiti	35
3.2	Fasi preliminari di ogni progetto SAP	37
3.3	Customizing	38
3.3.1	Display Profile	38
3.3.2	Presentation Profile e Personalization Profile	38
3.3.3	Presentation Device	39
3.3.4	Menu Manager	39
3.4	Le interfacce	40
3.4.1	Scambio di informazioni tra interfacce	40
3.4.2	Prima Schermata	41
3.4.3	Seconda schermata	41
3.5	ABAP Dictionary	42
3.5.1	Dictionary Object	43
3.6	La transazione ZINSN	45
3.6.1	RF Framework Interaction Model	45
3.6.2	Steps	46
3.6.3	Function Code Profile	46
3.6.4	Architettura	47
3.6.5	Function Modules	49
3.7	Lock Object	52
3.8	Test	53
3.8.1	Fasi preliminari: Creazione di una Inbound Delivery	53
3.8.2	Dimostrazione dell'uso e del corretto funzionamento della transazione singolo utente	54
3.8.3	Dimostrazione dell'uso e del corretto funzionamento della transazione più utenti alla volta	58
3.8.4	Gestione degli Errori e Warning	59
	Conclusioni	61
	Ringraziamenti	63
	Bibliografia	65

Elenco delle figure

1.1	Architettura SAP a tre livelli	8
1.2	Transport system	10
1.3	Le transazioni SAP	11
1.4	Unità organizzative SAP-EWM	14
1.5	Architettura del RF Environment	21
2.1	Processo di ricezione merce in WM	24
2.2	Movimenti multi-step in EWM	25
2.3	Comunicazione tra sistemi ERP ed EWM	26
2.4	Processo entrata merci in WM ed EWM	28
2.5	Flusso del processo in base ai requisiti dei numeri seriali	29
2.6	Gestione dei numeri seriali tra WM ed EWM	31
2.7	Inserimento dei numeri seriali in una inbound delivery	32
2.8	Schema inserimento manuale dei numeri seriali in una inbound delivery	32
2.9	Schema inserimento dei numeri seriali in una inbound delivery attraverso un terminale in RF	33
3.1	Mockup prima schermata	35
3.2	Mockup seconda schermata	36
3.3	Struttura unità di progetto	37
3.4	Definizione delle voci del Menu Manager	39
3.5	Scambio di informazioni tra schermate RF	40
3.6	Screen Painter della schermata 001	41
3.7	Screen Painter schermata 002	41
3.8	ABAP Dictionary	42
3.9	Struttura ZCS_SN_MM	43
3.10	Struttura ZCS_SN_MATERIAL_MM	44
3.11	Tabella ZCT_SN_MATERIAL_MM	44
3.12	Modello di interazione del Framework RF	45
3.13	Definizione degli step della transazione ZINSN	46
3.14	Diagramma degli stati della transazione ZINSN	47
3.15	Step Flow di ZINSN	47
3.16	Flusso PAI/PBO	48
3.17	Transazione MIGO	53

3.18	Serial Numbers nella Inbound Delivery, transazione /SCWM/- PRDI	54
3.19	Percorso Menu transazione ZINSN	55
3.20	Prima Schermata: inserimento Inbound Delivery	56
3.21	Seconda Schermata: inserimento Serial Numbers	56
3.22	Inserimento Seriale DL333SN777414	57
3.23	Save, Back e reinserimento Inbound Delivery 6103	57
3.24	Schermata inserimento seriali: Utente A e Utente B	58
3.25	Schermata Utente B dopo il salvataggio	59
3.26	Schermata Utente A dopo il secondo salvataggio	59

Introduzione

Negli ultimi anni se un'azienda vuole essere competitiva sul mercato globale è sempre più fondamentale avere un'organizzazione interfunzionale, in cui la gestione per processi incoraggi un coinvolgimento nel business di tutte le funzioni aziendali attraversate trasversalmente dai processi. Affinché questo possa avvenire con successo, è necessario che nell'azienda i dati e le informazioni siano condivisibili da tutti coloro che ne possano aver bisogno: è indispensabile quindi una completa integrazione gestionale. È in quest'ottica che un sistema informativo diventa una leva strategica per migliorare la qualità dei flussi di dati e d'informazioni: ottimizzando i processi e permettendo operazioni di Business Process Reengineering (BPR). Fino a qualche anno fa, le aziende erano obbligate a crearsi il software informativo su misura a seconda delle proprie esigenze, commissionando l'opera a delle software houses. Oggi invece è prevalente il caso dell'acquisto di pacchetti **ERP** (Enterprise Resource Planning) che definiscono a priori le logiche operative da utilizzare sulla base delle migliori pratiche disponibili nell'ambito operativo ma comunque adattabili alle specifiche esigenze dell'azienda attraverso la loro parametrizzazione. Pertanto con l'aumento della popolarità dell'ERP e la riduzione dei costi per l'ICT (Information and Communication Technology), si sono sviluppate applicazioni che aiutano i business manager ad implementare questa metodologia nelle attività di business come: controllo di inventari, tracciamento degli ordini, servizi per i clienti, finanza e risorse umane. Il software gestionale ERP leader del mercato è sicuramente **SAP**, il quale possiede una suite di applicazioni integrate in tutti i settori industriali. SAP prevede vari moduli, a seconda della funzione aziendale a cui sono rivolti. Di conseguenza sono nate numerose aziende specializzate nella consulenza e nella personalizzazione della piattaforma ERP sviluppata da SAP in base alle esigenze specifiche del cliente. Una di queste aziende è **Engineering Ingegneria Informatica s.p.a.** che, nell'ambito operativo di una delle sue divisioni aziendali, si occupa dell'implementazione di questo software, e di alcuni suoi moduli, su richiesta di partner di grande rilevanza nel mercato. Uno dei moduli più utilizzati in ambienti come grandi impianti di stoccaggio o siti di produzione con annessi magazzini per la gestione dei processi di trasporto in entrata e in uscita oltretutto i movimenti all'interno di esso, è **SAP-EWM** (Extended Warehouse Management). SAP EWM offre varie funzionalità tra le quali l'utilizzo di transazioni in radiofrequenza che non sono

altro che interfacce utente semplici e altamente specializzate che supportano l'operatore di magazzino nelle attività. Queste transazioni RF sono eseguite su dispositivi portatili con piccoli schermi come palmari oppure su dispositivi montati su un carrello elevatore con uno schermo più grande. Tra le circa 1600 transazioni RF manca però una transazione standard che permetta all'operatore di inserire i **numeri seriali** dei prodotti in arrivo al magazzino potendo quindi gestire i numeri di serie a livello della delivery. Salvare a sistema sin dall'arrivo merci i seriali dei prodotti può essere molto importante perché rende possibile tener traccia dell'intero ciclo di vita all'interno del magazzino dei prodotti serializzati. In particolare è possibile fare un controllo dello stock anche ai fini della garanzia e capire quando quel determinato prodotto è stato acquistato, da chi e tutte le altre informazioni salvate nell'inbound delivery. In questa tesi, sviluppata all'interno dell'azienda Engineering Ingegneria Informatica, viene progettata ed implementata quindi una transazione in radiofrequenza che permetta all'utente di non esser costretto a svolgere un compito stazionario dovendo utilizzare una transazione standard attraverso un PC fisso ma poter sfruttare il palmare a sua disposizione per scannerizzare in maniera molto più veloce e accurata i codici seriali dei prodotti in fase di ricezione merce. Questa transazione una volta implementata è stata testata su dati reali di un sistema EWM di una nota azienda di macchine di caffè. In particolare sono state testate tutte le casistiche che possono accadere durante l'utilizzo di questa transazione da parte di un magazziniere, gestendo tutti gli errori ed andando in aiuto all'utente mostrandogli messaggi specifici. Lo scopo però è stato quello di realizzare una funzionalità in aggiunta a quelle che lo standard SAP propone, che possa funzionare per una qualsiasi azienda che adotta il modulo SAP-EWM, ma che comunque prima sia stata testata in un database reale. L'unico cambiamento nel caso in cui questa transazione voglia essere utilizzata da un'azienda diversa da quella per cui si è testata è quello di cambiare semplicemente la parte di gestione del seriale in base alla propria strategia. La tesi è strutturata dunque in un **primo capitolo** dove viene presentata una panoramica sulle tecnologie e sul dominio applicativo del progetto sviluppato. In questo capitolo vengono pertanto esposte le caratteristiche principali dei sistemi ERP e del relativo software gestionale SAP, analizzando specificatamente il modulo SAP-EWM riguardante la gestione dei magazzini. Qui sono stati evidenziati anche gli aspetti fondamentali dell'ambiente di sviluppo: il linguaggio di programmazione ABAP e il framework RF per le transazioni in esecuzione nei dispositivi portatili. Nel **secondo capitolo** viene analizzato il processo di magazzino nel punto in cui si va ad aggiungere una funzionalità non presente dallo standard in base alle esigenze che può avere una azienda che utilizza il modulo SAP-EWM. Precisamente viene approfondito il processo che riguarda la fase di ricezione della merce, mostrando prima quello che è il flusso standard, per poi illustrare i benefici che porta la nuova implementazione fissando gli obiettivi da raggiungere. Il **terzo capitolo** presenta il progetto nella sua interezza. Nella prima parte viene effettuata l'analisi dei requisiti per

poi procedere con l'approfondimento delle operazioni di personalizzazione dell'ambiente RF con cui interagiscono gli operatori di magazzino attraverso una fase di customizing del sistema. Nella seconda parte viene descritta sommariamente la parte di implementazione, mentre nell'ultima viene esposta una fase di test per verificare il corretto funzionamento della transazione. Infine, nel **capitolo conclusivo**, si evidenziano gli obiettivi raggiunti in questo progetto di tesi e alcune caratteristiche del prodotto finale oltrechè vengono esposti eventuali sviluppi futuri.

Capitolo 1

Le tecnologie

In questo primo capitolo si vuole fornire una panoramica sulle tecnologie e sul dominio applicativo del progetto sviluppato.

L'obiettivo è quello di permettere al lettore di acquisire le nozioni riguardanti gli argomenti trattati nei capitoli successivi, in particolare le caratteristiche della piattaforma ERP di SAP e del suo modulo SAP-EWM, oltreché dell'ambiente di sviluppo.

1.1 Sistema ERP

ERP significa **Enterprise Resource Planning** (“ pianificazione delle risorse d'impresa ”). Si tratta di un sistema di gestione che integra tutti i processi di business rilevanti di un'azienda.

Un sistema ERP possiede tre caratteristiche principali che rispecchiano una precisa concezione del sistema informativo:

- **Unicità dell'informazione:** un database comune per tutte le applicazioni che porta numerosi vantaggi:
 - sincronizzazione dei dati che permette la sincronizzazione dei processi interdipendenti;
 - assenza di ridondanza senza la necessità di complesse procedure di aggiornamento dei dati nelle diverse isole aziendali;
 - tracciabilità degli aggiornamenti, individuando perchè e chi ha modificato un dato valore;
 - affidabilità dell'informazione in quanto provenendo da un'unica sorgente non si rischia di avere inconsistenza o perdita di credibilità.
- **Estensione e modularità:** si ha la possibilità di scegliere l'approccio d'acquisizione dei moduli più consono all'azienda. Questo permette di poter diluire il pagamento del software e la formazione necessaria per il personale concentrandosi su un modulo alla volta;

- **Un approccio prescrittivo:** il sistema ERP definisce la logica operativa secondo le best-practice con la possibilità però di parametrizzare. I sistemi ERP garantiscono quindi la correttezza e la standardizzazione delle operazioni, oltre che a favorire la razionalizzazione dei processi, facendo coincidere il progetto informatico ERP con un progetto di Business Process Reengineering (BPR).

1.1.1 Vantaggi e svantaggi dei sistemi ERP

L'utilizzo di un software gestionale ERP porta con sé numerosi vantaggi dovuti principalmente alla centralizzazione dei dati in un unico database. In particolare permette di:

- Possedere una visione globale dei processi aziendali;
- Avere un controllo totale del business core;
- Sincronizzare esattamente le modifiche tra i diversi sistemi, così come tra tutti i reparti aziendali coinvolti;
- Avere una scalabilità delle informazioni che sono sempre disponibili in qualsiasi momento anche attraverso dispositivi quali smartphone e tablet;
- Supportare i manager nel prendere decisioni appropriate supportate da dati ed informazioni;
- Conferire un elevato grado di sicurezza riducendo anche notevolmente la possibile perdita di dati sensibili.

I sistemi ERP portano però anche ad alcuni svantaggi:

- Costi elevati che li rendono più adeguati solo per le medio-grandi aziende;
- Rigidità di adattamento rispetto ai flussi di lavoro dovendo far aderire i processi aziendali a quelli definiti nell'ERP;
- Tempi prolungati per i processi di integrazione dei sistemi ERP all'interno del contesto aziendale.

Questi sono i principali vantaggi e svantaggi relativi all'utilizzo di software gestionali di ERP. I vantaggi sono notevoli e i limiti sono dovuti principalmente a fattori riguardanti l'utilizzo da parte degli operatori o costi e integrabilità degli strumenti stessi. Risultano pertanto evidenti le potenzialità e le funzionalità, tanto quanto l'importanza a livello di investimenti economici di formazione e di integrazione all'interno dei processi aziendali.

1.2 SAP

Il software gestionale ERP leader del mercato è sicuramente **SAP**, il quale viene utilizzato da centinaia di migliaia di clienti, il quale possiede una suite di applicazioni integrate in tutti i settori industriali. Questo software è proprietà della multinazionale SAP-SE, che nasce nel 1972 in Germania ad opera di cinque dipendenti della IBM, Claus Wellenreuther, Hans-Werner Hector, Klaus Tschira, Dietmar Hopp e Hasso Plattner, dove svilupparono il primo sistema R/1. Questo sistema è stato utilizzato per la contabilità finanziaria e non includeva i livelli di presentazione, applicazione o database che si trovano oggi nel sistema SAP. Alla fine degli anni '70, SAP introdusse il sistema R/2, nel quale vi erano due livelli: livello di presentazione e applicazione ed in più il livello del database. Con l'evolversi della tecnologia, SAP si spostò verso un'architettura a tre livelli con l'introduzione di SAP R/3 negli anni '90. Il sistema SAP R/3 comprende quindi questi tre livelli: il livello di presentazione, il livello dell'applicazione e il livello del database. Nei primi anni del 2000 avviene la più recente evoluzione dei sistemi SAP in risposta alla diffusione delle tecnologie Internet. Questa nuova versione aggiunge il supporto all'accesso al sistema tramite browser Internet e sistemi basati su tecnologia Java grazie ad un Web server proprietario in grado di offrire un'interfaccia di comunicazione comune per lo scambio di informazioni con sistemi R/2, R/3 e di terze parti. La nuova piattaforma venne denominata SAP NetWeaver e il nuovo componente centrale del sistema, in sostituzione del precedente SAP Basis, venne chiamato SAP NetWeaver Application Server. [2]

Anno	Versione
1972	SAP R/1
Fine anni '70	SAP R/2
1992	SAP R/3
2001	Introduzione di SAP Web Application Server
2002	SAP R/3 Enterprise 4.70 (SAP Web Application Server 6.20)
2003	mySAP ERP 2003 (SAP Web Application Server 6.30)
2004	mySAP ERP 2004 ECC 5.0 ERP 5.0 (SAP Web Application Server 6.40; SAP NetWeaver 2004)
2005	mySAP ERP 2005 ECC 6.0 ERP 6.0 (SAP Web Application Server 7.00; NetWeaver 2004s/7)
2006	SAP NetWeaver 7.1
2011	SAP NetWeaver 7.3
2013	SAP NetWeaver 7.4
2015	SAP NetWeaver 7.5

Tabella 1.1: Tabella Versioni SAP

Uno dei motivi per cui SAP è popolare deriva dalla sua flessibilità e dalla sua facile personalizzazione. Per ottenere questa elasticità, un sistema SAP è diviso in vari moduli. Ogni **modulo SAP** si associa a un processo aziendale di una business unit. Per esempio il modulo SAP ERP MM si associa all'attività di gestione dei materiali, il modulo HR si associa all'unità organizzativa delle risorse umane, il modulo WM si associa all'attività di gestione del magazzino, e così via. Tutti i moduli SAP possono essere integrati tra loro o con sistemi di terze parti. In base all'organizzazione, è possibile implementare uno o più moduli SAP e integrarli con altri sistemi ERP.

1.2.1 Architettura SAP

A partire quindi dalla versione SAP R/3 il sistema è strutturato in un'architettura a **tre livelli**, il numero 3 sta a significare proprio questo mentre R sta a significare **real-time** (scambio di dati istantaneo). L'architettura a tre livelli è un'architettura **client-server** in cui l'interfaccia utente (**Presentation layer**), la business logic (**Application layer**) e l'archiviazione dati (**Database layer**) sono sviluppati e mantenuti come moduli indipendenti (spesso su piattaforme separate).

Come si evince dalla figura 1.1, il livello di presentazione comunica con il livello dell'applicazione e il livello dell'applicazione comunica con il livello del database. Il livello di presentazione non può comunicare direttamente con il livello del database, il che migliora ulteriormente la sicurezza per logica aziendale, perché il client non ha accesso diretto al database.

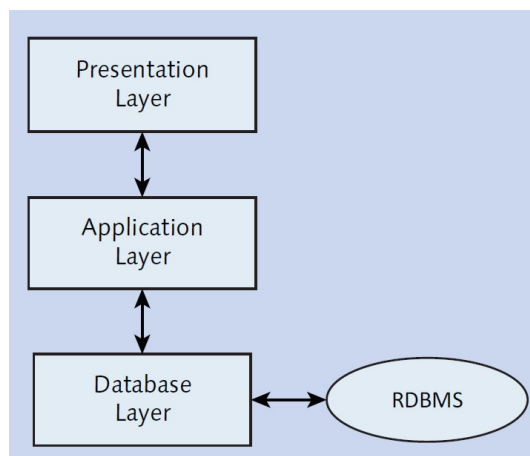


Figura 1.1: Architettura SAP a tre livelli

In modo più approfondito:

- **Presentation layer:** il livello di presentazione è costituito dall'interfaccia utente, che viene eseguita su un computer client che utilizza un'interfaccia utente grafica standard (GUI). L'interfaccia utente può essere un browser Web o un software di proprietà come SAP GUI. Lo strato

di presentazione può essere sviluppato indipendentemente in base ai requisiti e configurazione del sistema, può supportare vari dispositivi (PC, smartphone, ecc.) e sistemi operativi. La GUI comunica con il livello dell'applicazione prendendo gli input dell'utente, inviando richieste al server e visualizzando i risultati ricevuti dal server alle applicazioni.

- **Application layer:** il livello dell'applicazione è il luogo in cui tutto il lavoro principale viene svolto dai programmi. Questo è il livello software che ospita vari moduli di funzioni aziendali ed esegue il codice del programma. Può essere costituito da uno o più server di applicazioni ed un server di messaggistica, responsabile della comunicazione all'interno del livello. Quando si esegue un'applicazione, l'esecuzione avviene nel livello applicazione. Il livello dell'applicazione può comunicare con il database per elaborare i dati richiesti dall'utente.
- **Database layer:** il livello del database è dove vengono memorizzati tutti i dati persistenti. Il database è generalmente gestito da un sistema di gestione di database relazionali (RDBMS). RDBMS fornisce un metodo dichiarativo per conservare dati e query ed il sistema si occupa di descrivere le strutture per la memorizzazione dei dati. In un modello RDBMS, i dati sono memorizzati in una o più tabelle. Ogni tabella è composta da righe e colonne, ciascuna riga identificata da una chiave univoca. Le tabelle sono correlate tra loro con chiavi esterne. Se il campo chiave di una tabella esiste come campo non chiave in un'altra tabella, le due tabelle hanno una relazione di chiave esterna. Le chiavi esterne consentono di definire una relazione genitore-figlio tra diverse tabelle del database. SAP supporta vari sistemi di database, come SAP HANA, Oracle, MSSQL, MaxDB e IBM DB/2.

1.2.2 Tipologie di sistemi SAP

Per quanto riguarda l'organizzazione di un progetto sviluppato con SAP, essa prevede nel caso generale tre tipologie di sistemi:

- **Test and Development system:** in cui sono presenti i dati di sviluppo. I settaggi di customizing vengono mantenuti inizialmente solo in questo sistema, dove vengono testati e ottimizzati in base delle specifiche di progetto definite con il cliente finale.
- **Consolidation (or Quality Assurance) system:** qui vengono trasportati gli oggetti sviluppati e testati sul Development system per verificarne la qualità prima di eseguire il trasporto verso il sistema di produzione. Il sistema di Quality ha dunque il compito di verificare e assicurare che lo sviluppo effettuato rispetti i requisiti di qualità.

- **Production system:** dopo essere stati testati e rilasciati, i dati vengono trasportati verso il Production system, il quale presenta il livello di protezione dei dati più alto fra tutti i sistemi. I parametri di progetto non possono essere direttamente modificati su questo sistema, a differenza degli altri sistemi dove era invece possibile.

La configurazione minima di un progetto reale richiede due sistemi, Test e Production, entrambi in esecuzione sulla stessa o su diverse macchine fisiche a seconda del sistema operativo e dell'hardware presente sul server.

L'unico punto di contatto tra i differenti sistemi è rappresentato da un ulteriore componente, il **Transport system**, il cui compito è quello di trasferire le modifiche effettuate su un sistema al sistema successivo mediante l'utilizzo di pacchetti chiamati **Change Request (CR)** o **Transport Request** (Fig. 1.2).



Figura 1.2: Transport system

Ogni CR contiene almeno uno (o più) **Change Jobs**, chiamati anche **Change Tasks**, ovvero le minime unità trasportabili. Ogni Task è assegnato ad un unico utente, responsabile dei cambiamenti, e contiene al suo interno una lista di oggetti da rilasciare (non modificabili in contemporanea da altri utenti in altri task). Ad ogni CR possono essere assegnati più utenti, assegnandogli task diversi. Il project manager (o il responsabile designato) può creare una CR, le assegna i membri sviluppatori definendo specifici task per ognuno di loro, ed è incaricato di rilasciare la CR. Gli utenti invece sono responsabili del rilascio dei singoli Task assegnati.

Le modifiche contenute all'interno di una Change Request possono essere classificate in due macrocategorie relativamente al contenuto della modifica stessa:

- **Workbench request:** sono modifiche condivise da tutti i sistemi. Appartengono a questa categoria oggetti di repository come Function Module Z, tabelle, o modifiche al codice SAP standard;
- **Customizing request:** modifiche che hanno effetto solo sul sistema specifico. Appartengono a questa categoria tutte le impostazioni di Customizing eseguite dalla transazione SPRO.

1.2.3 Transazioni SAP

I componenti SAP sono costituiti da moduli che a loro volta comprendono **transazioni** utilizzate per eseguire processi aziendali (Fig. 1.3).

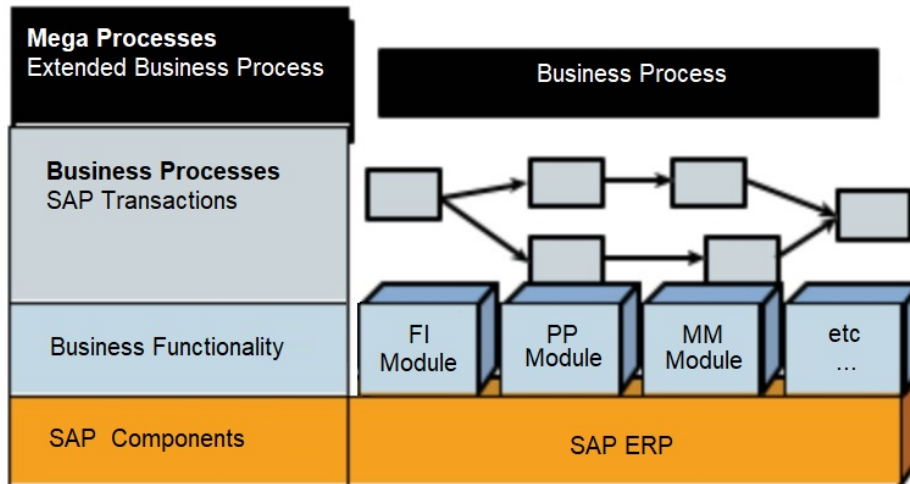


Figura 1.3: Le transazioni SAP

Queste transazioni sono moduli software con specifiche funzionalità messi a disposizione da SAP e destinati sia all'uso da parte di utenti per l'esecuzione di processi aziendali ma anche rivolti agli sviluppatori di programmi ABAP. Ogni transazione è identificata attraverso un Transaction Code ed è resa disponibile attraverso l'interfaccia SAP GUI. Di seguito sono riportate le descrizioni delle transazioni maggiormente utilizzate in fase di progetto [3]:

- **ABAP Dictionary (SE11)**: si utilizza l'ABAP Dictionary per creare e gestire le definizioni dei dati. Il dizionario ABAP consente una descrizione centrale di tutti i dati utilizzati nel sistema senza ridondanze. Le informazioni nuove o modificate vengono fornite automaticamente per tutti i componenti del sistema. Ciò garantisce l'integrità, la coerenza e la sicurezza dei dati. L'ABAP Dictionary supporta la definizione di tipi definiti dall'utente (elementi di dati, strutture e tipi di tabelle). È possibile creare gli oggetti corrispondenti (tabelle o viste) nel database relazionale sottostante utilizzando queste definizioni di dati. Il dizionario ABAP descrive la struttura logica degli oggetti utilizzati nello sviluppo dell'applicazione e mostra come sono mappati al database relazionale sottostante in tabelle o viste.
- **ABAP Editor (SE38)**: è uno strumento di modifica del codice sorgente e rappresenta l'editor di testo integrato all'interno della piattaforma. È uno dei componenti principali di ABAP Workbench. Si utilizza ABAP Editor per scrivere e modificare programmi ABAP, metodi delle classi, function module, logica di flusso degli screen, gruppi di tipi e database logici.

- **Function builder (SE37)**: è uno strumento in ABAP Workbench che consente di creare, testare e amministrare le function module in un ambiente integrato.
- **Object Navigator (SE80)**: è possibile utilizzare la transazione SE80 per organizzare la programmazione in un ambiente di sviluppo integrato. E' fondamentale nella gestione di grandi applicazioni composte da numerosi moduli, schermate ed oggetti di business logic. L'object navigator incorpora diversi browser utilizzati per vari scopi di sviluppo:
 - Repository Browser;
 - Repository Information System;
 - Transport Organizer;
 - MIME Repository;
 - Tag Browser;
 - Test Repository;
 - ABAP Unit Test Browser,

1.3 SAP-EWM

Uno dei moduli più utilizzati in ambienti come grandi impianti di stoccaggio o siti di produzione con annessi magazzini per la gestione dei processi di trasporto in entrata e in uscita, è **SAP Extended Warehouse Management**. Il modulo **EWM** fa parte della più ampia suite di applicativi chiamata Supply Chain Management (SCM). SAP EWM consente all'azienda di controllare i propri processi in entrata ed in uscita nel magazzino e i movimenti di merci all'interno di esso. Il processo in entrata riguarda lo stoccaggio delle merci e la loro posizione, mentre il processo in uscita implica il ritiro delle merci. Ogni volta che un materiale viene registrato in un magazzino, viene memorizzato nella storage bin e lo si può trovare in una posizione corrente. Con l'aiuto di SAP EWM tutti i movimenti merci sono controllati dal sistema e fornisce quindi gli strumenti per monitorare le attività e gestire in modo efficiente le scorte nel magazzino. È inoltre possibile gestire funzioni aggiuntive come la creazione di un numero di serie, numero di lotto, gestione del fornitore, ottimizzazione delle risorse e altri servizi dal valore aggiunto, consentendo dunque non solo di monitorare la quantità di merci nel magazzino ma di gestire altre funzioni critiche. Per quanto riguarda l'inventario EWM si occupa dunque non solo del conteggio della merce ma dello spostamento e del monitoraggio dell'ubicazione fisica delle merci registrati con i documenti specifici.

1.3.1 Caratteristiche principali EWM

Le caratteristiche principali del modulo EWM sono dunque:

- Configurazione del magazzino secondo le migliori pratiche implementate dalla piattaforma. Suddivisione dell'impianto tenendo conto anche della struttura fisica del magazzino garantendo che i percorsi dei materiali all'interno dell'edificio siano ottimali per un risparmio sia in termini di costi che di tempo;
- Utilizzando SAP EWM è possibile monitorare i processi in inbound e outbound (entrata e uscita dal magazzino) dei beni immagazzinati;
- È possibile eseguire il deconsolidamento delle handling unit (HU) che contengono prodotti diversi prima di riporli in diverse storage section;
- È possibile dal sistema EWM al sistema ERP: impostare un avviso per i dati modificati prima del ricevimento merci, annullare o correggere la ricevuta merci e una suddividere la consegna;
- Creazione automatizzata dei task operativi per le risorse umane dell'impianto, permettendo di ottimizzare i tempi e suddividere in modo efficiente il carico di lavoro;
- Consente di gestire e tracciare i veicoli e le altre unità di trasporto dal check-in al check-out, compresi i movimenti e altri compiti all'interno del cantiere;
- Comprende anche lo stoccaggio e la movimentazione di sostanze pericolose e il loro trasporto in conformità con le normative di SAP Environment Health e Safety (EHS);
- È possibile utilizzare il warehouse cockpit che consente di visualizzare graficamente le cifre chiave del magazzino e di valutare o monitorare le attività utilizzando tipi di grafici definiti;
- È possibile utilizzare il cross-docking che consente di eseguire il trasporto di HU attraverso diversi centri di distribuzione o magazzini fino a raggiungere la posizione finale nel magazzino.

1.3.2 Architettura EWM

L'architettura di SAP Extended Warehouse Management è caratterizzata da una varietà di componenti e oggetti applicativi che compongono la ricca funzionalità del sistema stesso.[4] In particolare:

- **Delivery processing:** il processo di consegna riceve richieste di archiviazione e recupero, nonché i requisiti per il trasferimento delle scorte,

che derivano dalla gestione degli ordini. È quindi strettamente legato al sistema ERP SAP e ai componenti “Warehousing” and “Transportation Management”;

- **Warehouse logistics:** la logistica del magazzino offre una molteplicità di funzioni necessarie per la pianificazione e l’esecuzione di attività all’interno del magazzino;
- **Inventory management:** la gestione dell’inventario consente la gestione delle scorte in diversi spazi a livello di spazio di archiviazione e unità di parcheggio. È strettamente collegato alla gestione dell’inventario di SAP ERP;
- **Quality management:** per il controllo di qualità delle merci in entrata, SAP EWM fornisce uno stretto collegamento al Quality Inspection Engine (QIE);
- **Integration with SAP ERP:** l’integrazione con SAP ERP può essere considerata una componente trasversale di SAP EWM.

1.3.3 Unità organizzative

Il sistema SAP prevede diverse unità organizzative in un sistema di magazzino a diversi livelli (Fig. 1.4):

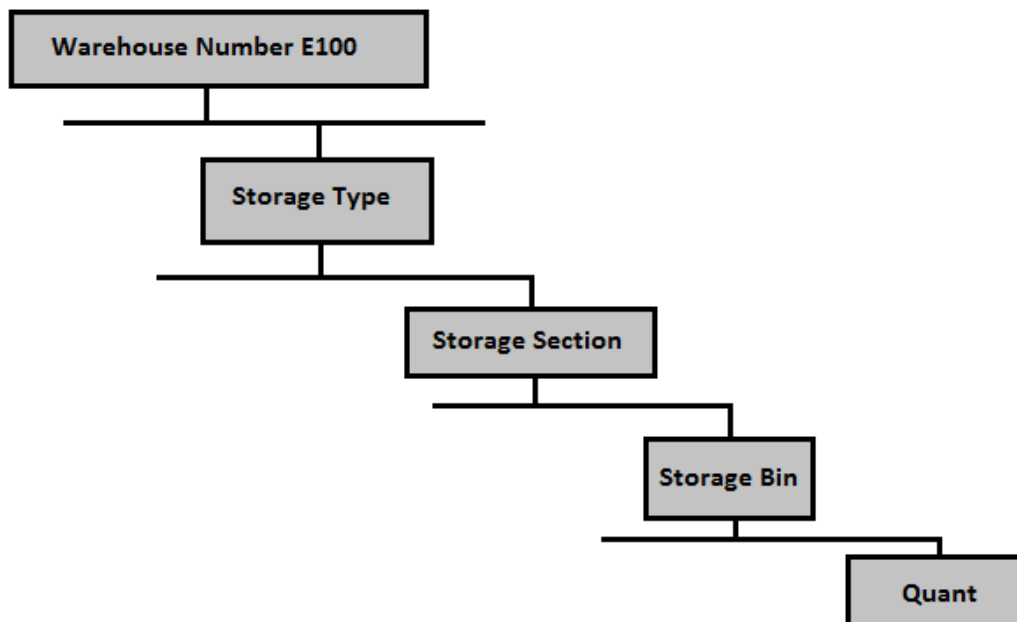


Figura 1.4: Unità organizzative SAP-EWM

- **Warehouse Number:** ciascun magazzino è costituito da un numero di magazzino che si trova al livello più alto in un sistema di gestione del magazzino. Ogni warehouse number è costituito da una sottostruttura;
- **Storage Type:** esistono diversi tipi di archiviazioni in cui i prodotti sono fisicamente conservati come open storage, goods issue e goods receipt, racks;
- **Storage Section:** questa è una parte della storage type e rappresenta un gruppo di contenitori con le stesse caratteristiche. Le sezioni di memoria comunemente utilizzate sono oggetti in rapido o lento movimento, ecc.;
- **Storage Bin:** rappresentano la posizione fisica dello spazio di archiviazione in cui i prodotti sono archiviati. Si trovano al livello più basso della struttura organizzativa e quando si memorizza un oggetto in un magazzino è necessario menzionarne la posizione esatta;
- **Activity Areas:** le storage bin sono ulteriormente suddivise in questa activity area. Include attività come la raccolta, l'archiviazione o l'inventario fisico. Secondo l'attività, è possibile assegnare lo stesso storage bin a più aree di attività;
- **Quant:** la quantità di merci nelle storage bin è rappresentata dal quant. Il quant è utilizzato per la gestione dell'inventario nelle storage bin.

1.4 Linguaggio di programmazione ABAP

All'interno dell'ambiente SAP è disponibile il linguaggio **ABAP/4** (acronimo di **Advanced Business Application Programming**) per lo sviluppo dei componenti software necessari per integrare SAP ad altri programmi di gestione. ABAP/4 è il linguaggio creato dalla SAP-SE per l'implementazione e lo sviluppo del sistema SAP/R3.

ABAP è un linguaggio di programmazione ibrido, questo perché nel tempo ha avuto una sua evoluzione in quanto è nato come linguaggio procedurale per poi essere esteso con i concetti della programmazione ad oggetti. Il linguaggio è sia compilato che interpretato, il codice infatti viene compilato da un compilatore integrato nella piattaforma in un formato intermedio per poi essere interpretato dall'ambiente di esecuzione ABAP in real-time.

Esso è un linguaggio che è stato creato appositamente per l'elaborazione massiva di dati in applicazioni commerciali, per questo motivo molte funzionalità avanzate sono direttamente incapsulate all'interno del linguaggio e non fornite da librerie esterne. Questo comporta un miglioramento delle performance ma anche un maggior numero di parole chiavi a disposizione del programmatore il quale dovrà valersi di un periodo d'apprendimento più lungo.

1.4.1 Tipologie di programmi

SAP distingue fra due tipologie differenti di programmi:

- **Report**: semplici programmi batch che generano in output liste, interfacce ecc.; l'utente non può intervenire durante l'esecuzione di un report per alterarne il comportamento;
- **Dynpros (Dynamic Programs)**: l'utente può intervenire durante l'esecuzione; questi programmi lavorano utilizzando una serie di dialog screens dove l'esecuzione del programma dipende da quali screen l'utente completa o da quali funzioni egli attiva.

Tutti i programmi girano sull'application server sotto forma di processi (**work processes**). In questo modo essi sono indipendenti dal sistema operativo e dal database sottostante.

1.4.2 Struttura di un programma

Ogni programma ABAP ha una struttura composta da due parti:

- **Declaration part**: definisce tabelle, tipi, strutture, aree di lavoro, campi, parametri per le selection screen ecc; i dati possono essere dichiarati come globali o locali a seconda delle necessità di utilizzo (dati dichiarati localmente, ad esempio all'interno di una procedura, saranno poi disponibili solo in tale procedura).
- **Processing blocks**: blocchi di codice che invocati secondo una determinata sequenza e che possono essere dei:
 - **Dialog modules**: chiamati dagli statement ABAP e sono ad esempio le selection screen o l'output visualizzato a video;
 - **Event blocks**: blocchi introdotti da parole chiave e che terminano implicitamente all'inizio di un nuovo evento o alla fine del programma;
 - **Procedure**: metodi, sottoprogrammi o function module che implementano la parte logica del programma.

1.4.3 Elementi del linguaggio

Definizione di dati e variabili

Il linguaggio ABAP/4 utilizza i **data type** per rappresentare i dati e le variabili. In particolare i più comuni sono:

Definizione	Data Type	Descrizione
Caratteri	c	Caratteri alfanumerici
	n	Caratteri numerici
Stringhe	string	Stringhe alfanumeriche
Numeri	i	Numeri in formato Integer
	p	Numeri in formato Packed
	f	Numeri in formato Floating point
Data	d	Data nel formato aaaammgg
Ora	t	Ora nel formato hhmmss
Esadecimale	x	Numeri esadecimali

Tabella 1.2: Tabella Data Type ABAP

Una variabile può essere definita attraverso una serie di parole chiavi che vanno dichiarate prima del dato nel codice sorgente:

- **DATA**: definisce una variabile in ogni contesto. È possibile utilizzare questa parola chiave nell'area di dichiarazione globale del programma ABAP per dichiarare una variabile globale che ha visibilità in tutto il programma, oppure è possibile utilizzarla all'interno di una procedura per dichiarare una variabile locale con visibilità solo all'interno della procedura;

Listing 1.1: Definizione DATA

```
1 DATA: amount TYPE f.
```

- **PARAMETERS**: questa parola chiave svolge un duplice ruolo. Definisce una variabile all'interno del contesto del programma e genera anche un campo dello schermo (selection screen). Questa parola chiave viene utilizzata per creare una selection screen per i programmi di report;

Listing 1.2: Definizione PARAMETERS

```
1 PARAMETERS: date TYPE d.
```

- **CONSTANTS**: viene utilizzata questa parola chiave per le costanti dove il contenuto non può essere modificato dalle istruzioni ABAP;

Listing 1.3: Definizione CONSTANTS

```
1 CONSTANTS: pi TYPE VALUE '3.14'.
```

- **FIELD-SYMBOLS**: campi definiti senza lunghezza e tipo di carattere, in grado di occupare una minima parte di memoria.

Listing 1.4: Definizione FIELD-SYMBOLS

```
1 FIELD-SYMBOLS: <amount> TYPE f.
```

Istruzioni per la logica di controllo

L'ABAP/4 è un linguaggio di tipo event-driven. Le azioni dell'utente e gli eventi del sistema controllano il flusso di una applicazione. Quindi come quasi tutti i programmi si ha bisogno di distinguere fra differenti situazioni a seguito di una condizione logica. Esistono due costrutti base per determinare quale tipologia di controllo applicare al programma:

- IF / ELSEIF / ELSE / ENDIF per una selezione a seguito di una condizione arbitraria;
- CASE / WHEN / ENDCASE per una selezione su di un insieme fisso di valori.

Funzioni

In un programma di grosse dimensioni spesso è necessario compiere più volte la stessa operazione. Per rendere più leggibile il programma, lo si può scomporre in moduli elementari che eseguono determinate operazioni; questi moduli potranno in seguito essere richiamati non solo dal programma in questione, ma altresì da programmi differenti.

In particolare le **functions** vengono normalmente chiamate da differenti programmi. Gli argomenti delle functions vengono identificati dai loro nomi e non dalla posizione che essi assumono. Se sorgono nuove richieste durante il ciclo vitale di una function, le si possono aggiungere come parametri opzionali senza dover cambiare tutte le calls ai programmi. Una function viene richiamata nella seguente sintassi:

```
CALL FUNCTION 'name' [EXPORTING par1 = var1]
[IMPORTING par2 = var2] [CHANGING par3 = var3]
[TABLES par4 = var4] [EXCEPTIONS exc = 1, others = 2]
[DESTINATION dest].
```

Per esempio:

Listing 1.5: Call Function Module

```
1 DATA: l_guid TYPE guid_32.
2 CALL FUNCTION 'GUID_CREATE'
3   IMPORTING
4   *   EV_GUID_16 =
5   *   EV_GUID_22 =
6   ev_guid_32 = l_guid.
```

Strutture e Tabelle interne

Una **struttura** raggruppa i dati in un programma ABAP, quindi è uno strumento per organizzare i dati in memoria e può anche aiutare a rendere più chiaro il codice del programma. Attraverso la type definition (parola chiave TYPES) ed utilizzando BEGIN OF / END OF è possibile raggruppare i dati in una

struttura e dichiararli tutti insieme. È quindi poi possibile accedere ad un singolo campo della struttura richiamando essa seguita da il trattino (-) e il nome del campo.

Listing 1.6: Structure

```

1  TYPES :
2      BEGIN OF name_structure ,
3          name(10) TYPE c ,
4          surname(10) TYPE c ,
5      END OF name_structure .
6
7  DATA :
8      variable_structure TYPE name_structure
9
10 variable_structure-name = 'Mattia' .
11 variable_structure-surname = 'Missere' .

```

A differenza delle tabelle di un database, le quali esistono sia che siano utilizzate o meno da un programma, le **internal tables** esistono soltanto per la durata di esecuzione del programma.

Listing 1.7: Internal Table

```

1  DATA :
2  internal_table TYPE TABLE OF name_structure .

```

Le internal tables possono essere viste come una fotografia della tabella del database che si desidera utilizzare all'interno del programma, in pratica sono un contenitore volatile di dati per il nostro programma. Una internal table fornisce la possibilità di operare con tutte le righe e le colonne della tabella contenuta nel database, oppure di escludere e/o aggiungere colonne e/o righe alla suddetta. Se nella definizione di una internal table viene l'aggiunta l'istruzione **WITH HEADER LINE**, il sistema fornisce sia oltre alla tabella stessa anche un data object aggiuntivo contenente una struttura come una riga individuale della tabella. Questo oggetto è utilizzato come record di default per contenere la riga corrente di una tabella. Se una tabella non ha header line, il programmatore deve fornire un record a parte per poter memorizzare il contenuto della riga corrente per quasi tutti i comandi che trattano con la tabella. In pratica la header line non è niente altro che un'area di lavoro, usata come memoria temporanea per un record di tabella del database.

ABAP ha tutta una serie di comandi per lavorare con le tabelle interne o con i record di queste ultime. Questi possono essere suddivisi in quattro aree:

- quella del riempimento (**INSERT**, **APPEND** e **COLLECT**);
- di lettura (**READ** e **LOOP**);
- di manipolazione (**MODIFY**);
- di cancellazione (**DELETE**).

Tabelle di Database

Oltre alla gestione dei dati in memoria mediante costrutti come strutture e tabelle interne, esistono meccanismi per l'accesso diretto ai dati memorizzati all'interno del database server. A differenza di altri linguaggi, dove per l'interazione con il database è necessario utilizzare un driver capace di stabilire una connessione e dialogare con il sistema di gestione delle tabelle, in ABAP è sufficiente utilizzare le parole chiave appositamente definite. Il linguaggio ABAP fornisce principalmente i seguenti metodi per manipolare le tabelle di database:

- leggere i dati all'interno del database utilizzando il comando **SELECT**;
- inserire nuove righe **INSERT**;
- cambiare i valori nelle righe della tabella **UPDATE**;
- inserire o cambiare i dati nelle tabelle di database **MODIFY**;
- cancellare record **DELETE**.

1.5 RF Environment

Le transazioni RF sono interfacce utente semplici e altamente specializzate. Durante l'esecuzione di un'attività fisica nel magazzino l'operatore utilizza un dispositivo mobile per confermare l'attività nel sistema. L'interfaccia utente RF fornirà all'utente informazioni precise per garantire che possa eseguire l'attività fisica passo dopo passo. SAP offre una raccolta di oltre ottanta transazioni RF, che possono essere utilizzate così come sono o utilizzate come modelli.

In un tipico progetto SAP EWM vengono utilizzati almeno due tipi di dispositivi RF: un dispositivo portatile con uno schermo piccolo e un dispositivo montato su un carrello elevatore con un grande schermo. La stessa transazione RF può essere eseguita su entrambi i dispositivi utilizzando la stessa logica aziendale ma diversi display. Oltre alla possibilità di usarlo per sviluppare nuove transazioni RF, il framework RF può offrire diverse caratteristiche: generare un report per regolare tutte le schermate standard sulle nuove dimensioni dello schermo; uno wizard per copiare e regolare singoli screens; la possibilità di definire menu RF per differenti gruppi di user.

1.5.1 Architettura del RF Environment

Le transazioni logiche sono sviluppate e mantenute nel RF framework e non possono essere eseguite come delle classiche transazioni SAP direttamente dalla schermata di SAP Easy Access (Fig. 1.5).

Le transazioni RF sono quindi in esecuzione nell'ambiente RF a partire dalla transazione `/SCWM/RFUI`.

Easy Enhancement Workbench (EEWB), che può essere utilizzato per gli arricchimenti del cliente, è integrato nel framework RF. All'interno dell'EEWB, gli utenti eseguono un processo RF e quando viene raggiunta la schermata di modifica, viene attivato lo strumento di miglioramento.[1]

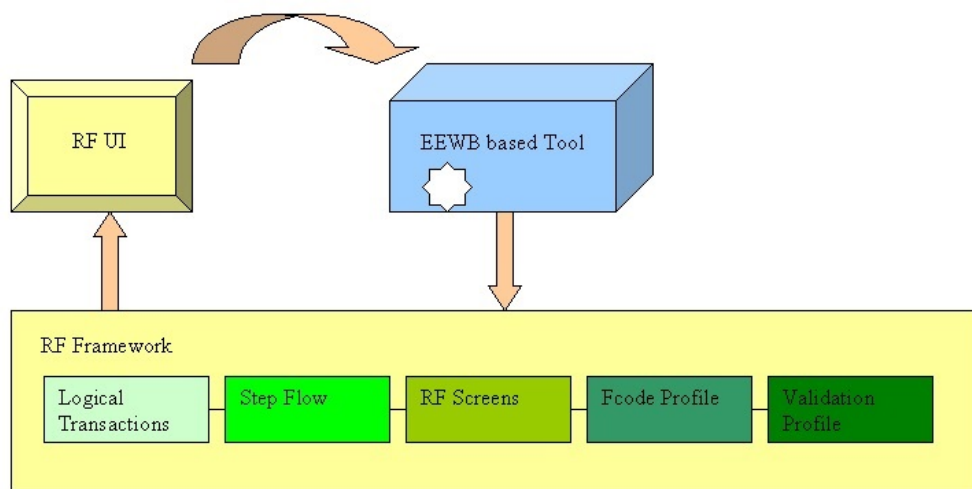


Figura 1.5: Architettura del RF Environment

L'architettura del RF framework è composta da tre livelli distinti:

- **Dynpro layer:** riguarda tutti i dati relativi allo schermo, i display profile, i presentation device, i subscreen, i template e così via;
- **Business logic layer:** in fase di runtime, il Business logic layer riceve i parametri transazionali;
- **Content provider:** memorizza e fornisce i dati rilevanti per l'elaborazione.

1.5.2 Definizioni fondamentali RF Environment

Di seguito vengono elencate le definizioni fondamentali per avere una comprensione delle transazioni logiche in radiofrequenza.[4]

- **Display Profile:** definisce le dimensioni dello schermo (generalmente in numero di righe e colonne) e il numero di pulsanti;

- **Application:** L'applicazione è l'unità organizzativa del framework RF. Inizialmente il framework è stato creato per ospitare applicazioni diverse da SAP EWM. Le transazioni RF di SAP EWM utilizzano l'applicazione 01;
- **Presentation Profile:** con esso è possibile specificare la logica specifica del magazzino in una transazione RF. Si collega ciascun magazzino ad esattamente un presentation profile, ma è possibile utilizzare lo stesso presentation profile in diversi magazzini;
- **Personalization Profile:** definisce, per un gruppo di utenti, quale set di transazioni RF sono disponibili e in quale struttura di menu sono elencate. Per un presentation profile, è possibile definire uno o più personalization profile;
- **RF Environment:** con la transazione /SCWM/RFUI si avvia l'ambiente RF in SAP GUI. La prima schermata è la schermata di accesso, in cui si specifica il magazzino, la risorsa e il presentation device. La seconda schermata è il menu principale in cui è possibile eseguire il drill-down in ulteriori sottomenu o transazioni RF. Ogni transazione RF può essere costituita da uno o più schermi;
- **Transazione logica RF:** è una finestra di dialogo con l'utente in esecuzione nell'ambiente RF. Ogni transazione RF viene avviata dal menu RF e solitamente consiste in uno o più schermi;
- **Step:** una transazione RF combina uno o più step. Di solito uno step consiste in uno schermo, diversi function code e diversi function module;
- **Application Parameter:** può essere una tabella o una struttura ABAP. I parametri dell'applicazione vengono trasferiti dal framework ai function modules all'interno di una transazione RF. Il framework aggiornerà l'application parameter con i valori immessi dall'utente sullo schermo e mostrerà i valori dei application parameter dopo che sono stati modificati nel function module;
- **Function Code:** può essere composto da un massimo di sei caratteri, associa un pulsante, fisico o virtuale, ad un function module della transazione capace di gestirlo.

Capitolo 2

Il processo di magazzino: ricezione merce

In questo secondo capitolo si analizza il processo di magazzino nel punto in cui si va ad aggiungere una funzionalità non presente dallo standard che può essere riutilizzata dalle aziende che richiedono questa esigenza. In particolare il processo riguarda il momento di ricezione della merce.

Lo scopo è quello di mostrare prima il flusso standard per poi illustrare i benefici della nuova implementazione andando ad inserire a sistema il numero seriale dei prodotti ricevuti attraverso una transazione in radio frequenza e non manualmente.

2.1 Soluzione WM per la ricezione merce

Nel sistema ERP WM esistono due metodi di base per l'elaborazione della merce in entrata:

- Il primo consiste nella pubblicazione della GR (goods receipt), seguito successivamente dalla put-away (riposizione) della merce in WM (ramo a sinistra della figura 2.1).
Lo stock in questo caso è già disponibile per le vendite mentre può essere fisicamente ancora nella zona di ricezione.
- Nel secondo caso (ramo a destra della figura 2.1) viene effettuata prima l'attività WM di put-away, seguita dalla GR in IM (Inventory Management).
Lo stock quindi diventa disponibile per le vendite solo dopo aver riposto effettivamente i prodotti.

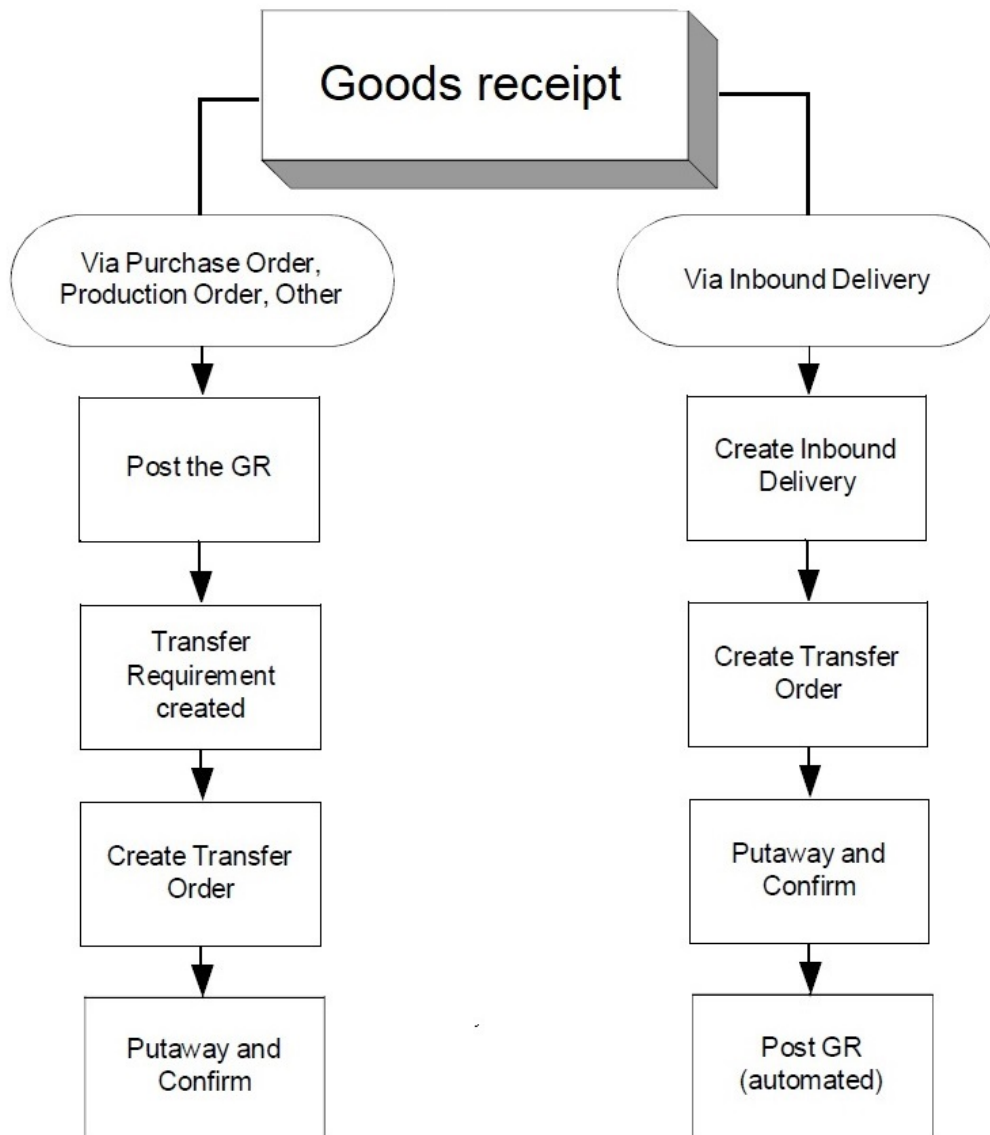


Figura 2.1: Processo di ricezione merce in WM

2.2 Movimenti multi-step di EWM

EWM si distingue fortemente da WM in particolare per una caratteristica (Fig. 2.2):

- WM si basa sul movimento a **passo singolo**. Un prodotto si sposta direttamente da un punto di origine ad una bin di destinazione (scopartimento finale). Questo movimento è controllato dal **warehouse movement type** (“tipo di movimento”).
- EWM può operare in modalità singola o **multi-step** per spostare un prodotto da un punto di origine ad una bin di destinazione. Per esempio:

- Prodotti che devono essere sottoposti ad ispezione di qualità di VAS;
- Prodotti che devono entrare in corridoi stretti: un primo passo li porta di fronte al corridoio, un secondo passo rappresenta il put-away dal carrello elevatore al corridoio stretto;
- Offloading in un'area di sosta, quindi rimessa negli scomparti finali.

Il termine warehouse movement type è sostituito in EWM dal termine **warehouse process type** (“tipo di processo”). Il warehouse process type ha anche un insieme ampliato di parametri di controllo rispetto a ERP WM. Uno di questi parametri è il **storage process type** (“tipo di processo di archiviazione”). Il storage process type raggruppa i passaggi di una catena di tipi di storage temporanei da seguire durante il processo di archiviazione. In questo modo una HU può passare da un tipo di storage temporaneo ad un altro, fino a raggiungere il tipo di storage finale. Un esempio di un tipo di storage temporaneo è il De-consolidation Work Center. Grazie a questa funzionalità di suddividere in vari processi l'archiviazione dei prodotti, si ha la possibilità di inserire a sistema in un determinato step il numero di serie dei prodotti.

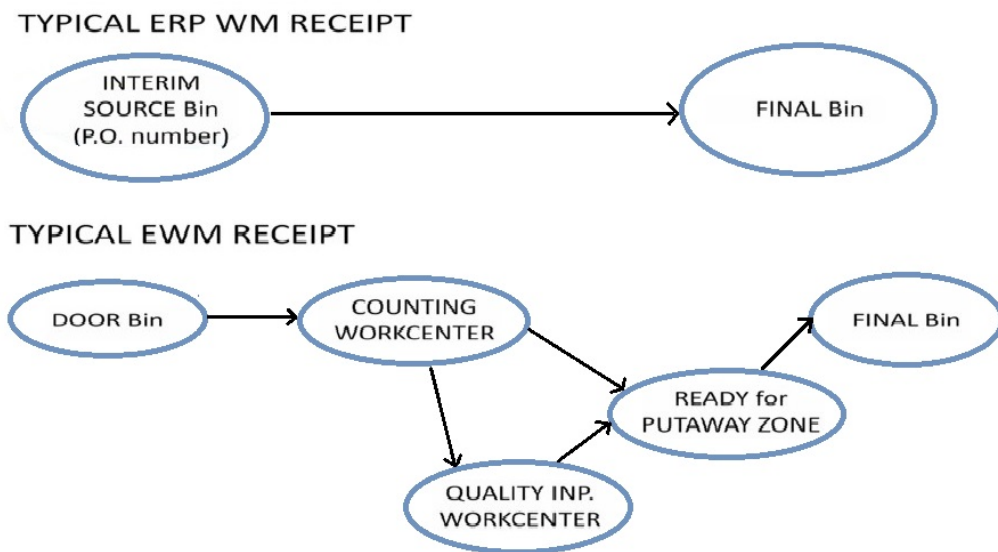


Figura 2.2: Movimenti multi-step in EWM

2.3 Comunicazione tra sistemi ERP ed EWM per le consegne

La comunicazione da Extended Warehouse Management (EWM) a SAP ERP avviene tramite interfacce definite, il che consente la distribuzione, le modifiche e la conferma dei dati rilevanti per la consegna. In SAP ERP, le interfacce tra EWM e SAP ERP si basano sulle **delivery**. In particolare in EWM, si distingue su quanto segue:

- **Inbound delivery notification** e **inbound delivery** nel processo di consegna in entrata;
- **Outbound delivery order** e **outbound delivery** nel processo di consegna in uscita.

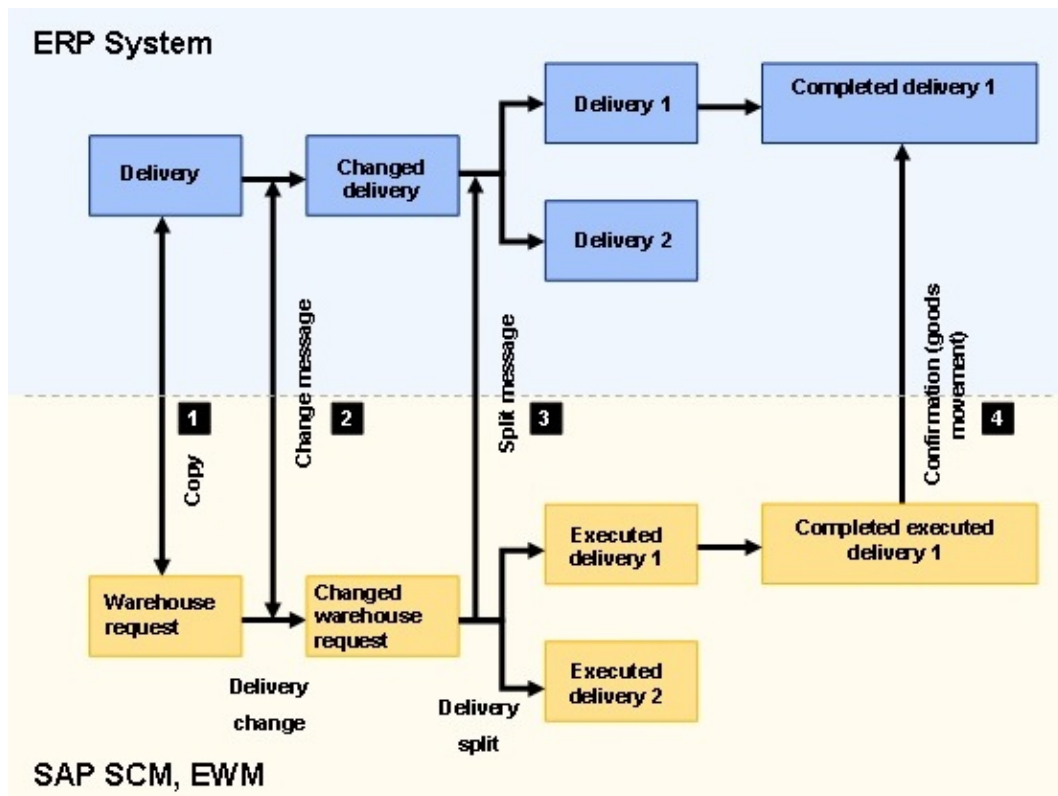


Figura 2.3: Comunicazione tra sistemi ERP ed EWM

La figura 2.3 offre una panoramica delle funzioni supportate dalle interfacce tra EWM e SAP ERP [3]:

1. Distribuzione delle consegne

- Inviare consegne da SAP ERP a EWM;

- Inviare consegne da EWM a SAP ERP (ad esempio, per consegne in entrata manuali).
2. Modifiche alle consegne
 - Inviare modifiche da EWM a SAP ERP (ad esempio, conferma di un rifiuto di prelievo da EWM a SAP ERP);
 - Invia modifiche da SAP ERP a EWM (ad esempio, modifica ASN dal fornitore).
 3. Confermare una consegna in entrata e in uscita da EWM a SAP ERP.
 4. Inviare la conferma delle consegne da EWM per le funzioni successive in SAP ERP (ad esempio, ricevimento merci, rilascio merce).

2.4 Soluzione EWM per la ricezione merce

Come è stato detto sopra, SAP utilizza la delivery come documento di comunicazione tra ERP ed EWM. In particolare per quanto riguarda le **inbound delivery** (documento di consegna in entrata) in ERP corrisponde in EWM due categorie di questi documenti:

- La **Inbound Delivery Notification (IDN)**, che è la copia della inbound delivery in ERP. Esiste una relazione uno-a-uno;
- La **Inbound Delivery (ID)** viene creata automaticamente in base all'IDN. L'ID è dunque la IDN arricchita con parametri di controllo come il Warehouse Process Type. L'ID è il documento di riferimento centrale per tutte le operazioni successive come la creazione di warehouse task, la GR, gli aggiornamenti di stato, ecc..

La figura 2.4 mostra il collegamento tra i diversi documenti durante il processo. Normalmente a seguito di un ordine di acquisto (purchase order) viene creata in ERP una inbound delivery ed invia al sistema EWM una inbound delivery notification. Lo scopo della notifica in EWM è quindi dividere in due categorie quello di evitare che il documento ERP rimanga bloccato in una coda. Infatti nel caso in cui mancassero dei dati, il sistema non sarà in grado di creare l'inbound delivery in EWM, ma riuscirà almeno a generare la inbound delivery notification. Come detto in precedenza per WM, anche per EWM è possibile effettuare la goods receipt in due diversi passaggi. Una soluzione è effettuare la GR appena la merce arriva in magazzino, l'altra è di effettuarla quando il prodotto arriva alla sua archiviazione finale dopo una serie di processi e movimentazioni. Un warehouse task può essere confrontato con un singolo Transfer Order in ERP. Il warehouse order (WO) è invece un raggruppamento di diverse attività di magazzino che devono essere eseguite da un operatore. Qui la grande differenza è che ERP WM non offre molto supporto transazionale

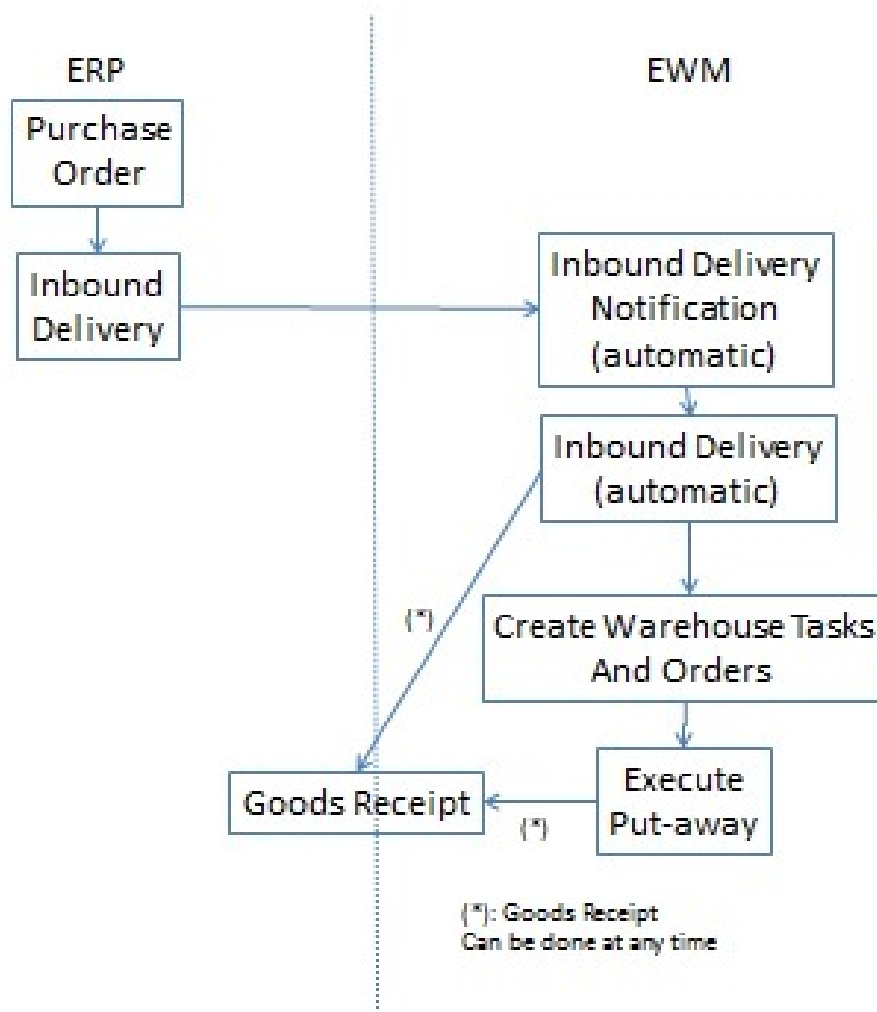


Figura 2.4: Processo entrata merci in WM ed EWM

per le attività in radio frequenza, né una reale logica di ottimizzazione su come realizzarli rispetto ad EWM. Proprio grazie a questo tipo di supporto e funzionalità che si potrà aggiungere l'attività di serializzazione dei prodotti nell'inbound delivery.

Dopo la conferma di un warehouse task, lo stato della consegna in entrata di EWM viene aggiornato e, a seconda di quando si verifica l'entrata merci, viene inviato un messaggio al sistema ERP per aggiornare la consegna in entrata (GR).

2.5 Serial Number Profile

Il **serial number profile** di SAP è un numero di quattro cifre in base al quale vengono definite le condizioni di funzionamento per l'assegnazione dei numeri di serie. Nella configurazione del sistema ERP, si definisce un serial number profile valido per tutti i numeri di magazzino. In EWM è invece possibile

definire un serial number profile dipendente dal numero di magazzino. In questo modo, è possibile fare in modo che un prodotto richieda un numero di serie in un magazzino e non ne richieda uno in un altro. Esistono due diversi tipi di serial number profile che possono essere utilizzati:

- **Sales General Plant:** devono essere definiti per utilizzare i numeri di serie nei documenti di consegna per la ricezione o invio di merci;
- **LES Serial Number:** viene mantenuto nella vista dei dati anagrafici del prodotto.

In EWM è possibile specificare quattro requisiti riguardante l'uso dei numeri seriali (Fig. 2.5):

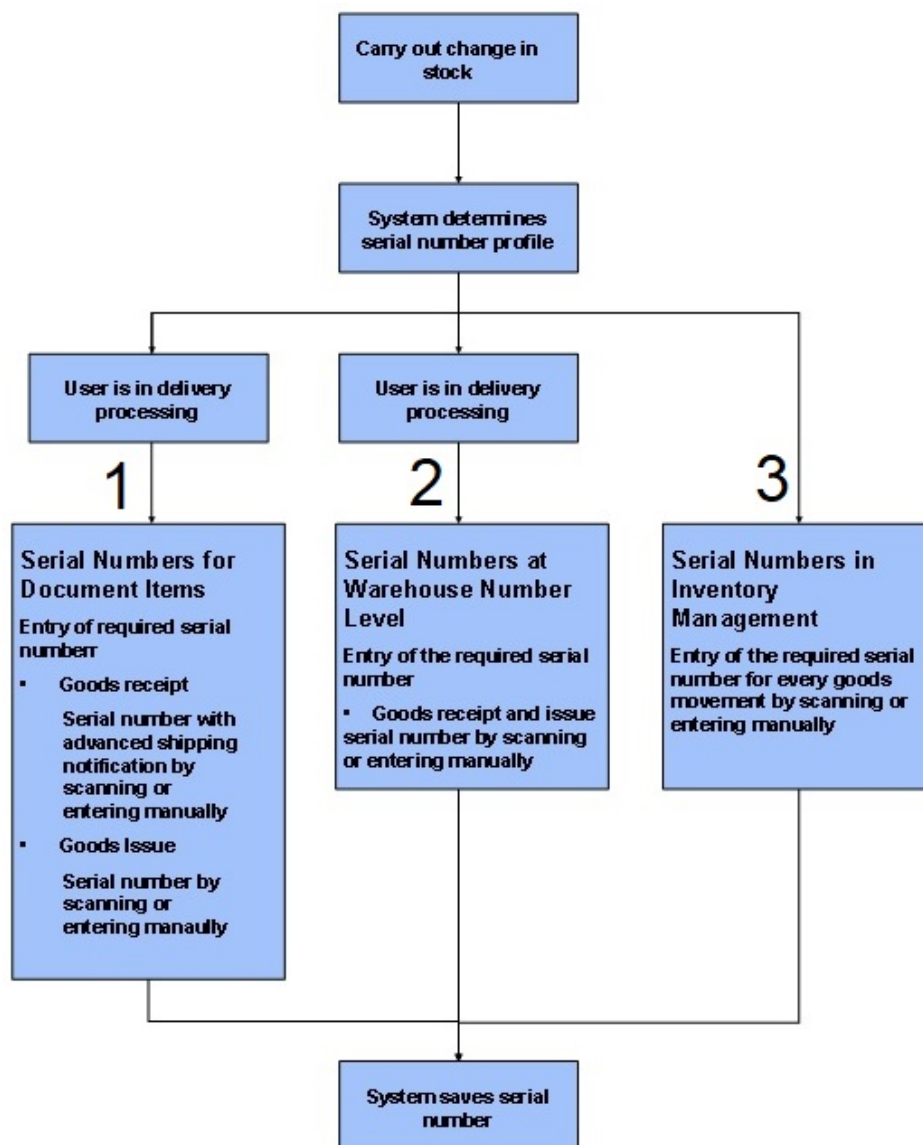


Figura 2.5: Flusso del processo in base ai requisiti dei numeri seriali

1. **Controllo a livello di item della delivery:** i numeri di serie vengono registrati solo per i prodotti nelle consegne in entrata o in uscita. Quando ad esempio un cliente si lamenta di un prodotto e viene richiesto un reso, è possibile eseguire un controllo se questo è lo stesso prodotto che gli viene fornito.
2. **Controllo a livello di magazzino:** i numeri di serie devono essere registrati in corrispondenza di ciascuna goods receipt e goods issue, potendo quindi conoscere quali numeri di serie sono nel magazzino (senza conoscere la loro posizione esatta).
3. **Controllo al livello di storage bin:** i numeri seriali vengono acquisiti per ogni conferma di attività nel magazzino. È possibile vedere quale numero seriale è esattamente dove.
4. **Nessun controllo sul numero di serie.**

È possibile utilizzare numeri seriali fino a una lunghezza di 30 caratteri in un sistema EWM. Tuttavia, SAP consiglia di utilizzare un massimo di 18 caratteri per renderlo compatibile con il sistema ERP.

2.6 Integrazione dei numeri seriali nelle consegne

Se nel customizing sono stati determinati i master data dei rispettivi prodotti in modo corretto, si possono assegnare i numeri di serie a ciascun articolo nella consegna. Inoltre non è possibile effettuare la buona riuscita dei movimenti se prima non vengono creati i numeri seriali dei prodotti che lo richiedono.

Per l'assegnazione del numero di serie, EWM apre una finestra di dialogo in cui è possibile inserire le voci desiderate. Questa attività però non è supportata da una transazione in RF.

Nel processo di ricezione merci, EWM può copiare il numero di serie degli articoli del documento dal sistema ERP. Nel fare questo, controlla che i dati per i numeri di serie non abbiano errori (Fig. 2.6):

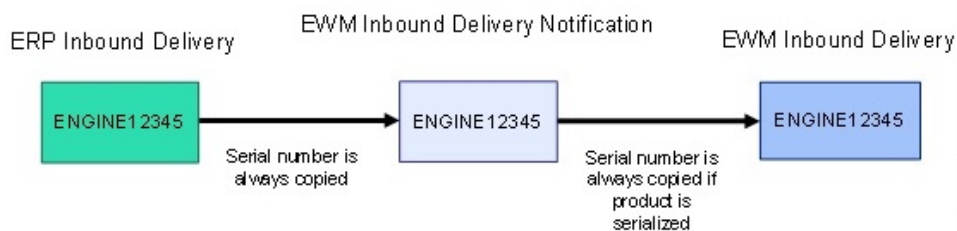
- Se non vi sono errori, EWM copia sempre i numeri seriali nell'inbound delivery notification.
- Se vi è un qualche tipo di errore come ad esempio se nel documento vi sono numeri di serie uguali per articoli differenti, EWM copia questi dati nella inbound delivery notification, ma poi non è possibile attivare la inbound delivery corrispondente. È possibile visualizzare il documento di consegna inattivo, scegliere la modalità di modifica e attivare il documento dopo aver corretto gli errori.

Una volta che EWM ha copiato i dati dal sistema ERP alla IDN, esegue una delle seguenti attività possibili:

- Se è stata determinata la serializzazione per il prodotto, EWM copia il numero di serie dalla inbound delivery notification alla corrispondente inbound delivery.
- Se non è stata determinata una serializzazione per il prodotto, EWM non copia il numero di serie nella inbound delivery.

Goods Receipt Process

Instance 1 – No errors in data from ERP system



Instance 2 – Data from ERP system contains errors

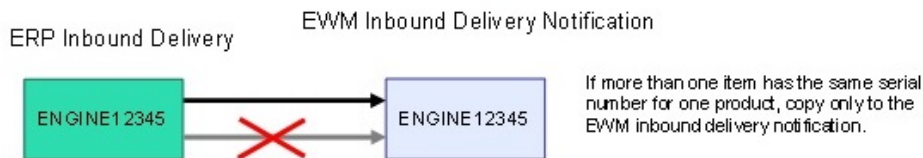


Figura 2.6: Gestione dei numeri seriali tra WM ed EWM

2.7 Inserimento dei numeri seriali a sistema a livello di item della delivery

Utilizzando i numeri di serie è possibile identificare i dettagli del fornitore e del destinatario che ha inviato e ricevuto merci in un magazzino.

2.7.1 Processo as is

In EWM è possibile inserire i numeri seriali nel documento di consegna attraverso la transazione **/SCWM/PRDI** (Fig. 2.7).

Inserendo l'ID dell'inbound delivery creata su sistema ERP si potrà visualizzare e modificare il documento in EWM.

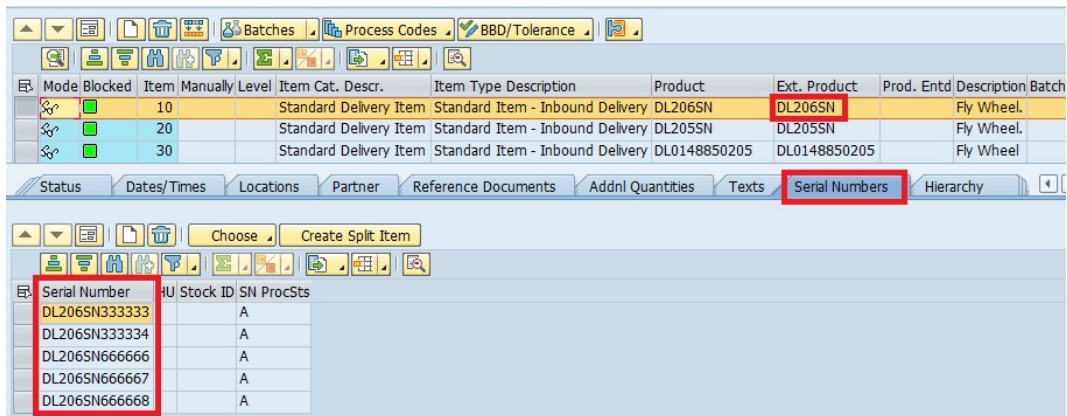


Figura 2.7: Inserimento dei numeri seriali in una inbound delivery

Dunque è in questa transazione che un utente dovrebbe inserire manualmente i serial number per ogni prodotto gestito a seriali di ogni item per poter poi eseguire la goods receipt.

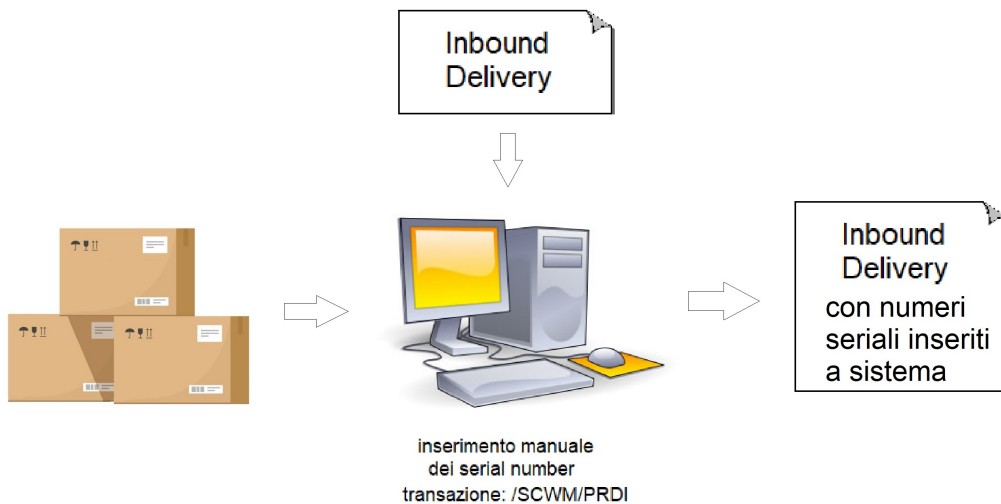


Figura 2.8: Schema inserimento manuale dei numeri seriali in una inbound delivery

Questo comporta un lavoro complesso e specialmente molto lungo (Fig. 2.8), è per questo motivo che entra in gioco la transazione custom non presente dallo standard per inserire tutti i numeri seriali attraverso la scannerizzazione con un dispositivo a radio frequenza.

2.7.2 Processo to be

La creazione di una transazione non presente nello standard può portare con sé una serie di numerosi vantaggi.

In questa maniera il magazziniere all'arrivo della merce potrà scannerizzare tutti i numeri seriali anche durante lo scarico dal camion (Fig. 2.9).

Obiettivi e vantaggi

Gli obiettivi e i relativi vantaggi di questa nuova funzionalità saranno dunque:

- Ridurre i tempi di inserimento di seriali a sistema dei prodotti in arrivo merci;
- Ridurre gli errori umani nella trascrizione dei numeri seriali;
- Permettere a più persone di lavorare e modificare una inbound delivery in contemporaneo e in maniera autonoma senza creare inconsistenza;
- Salvare a sistema sin dall'arrivo merci i seriali dei prodotti rende possibile tener traccia dell'intero ciclo di vita all'interno del magazzino dei prodotti serializzati. In particolare è possibile fare un controllo dello stock anche ai fini della garanzia e capire quando quel determinato prodotto è stato acquistato, da chi e tutte le altre informazioni salvate nell'inbound delivery.



Figura 2.9: Schema inserimento dei numeri seriali in una inbound delivery attraverso un terminale in RF

Problemi da risolvere

Questa però può portare anche una serie di costi e problemi che devono essere ben compresi:

- L'uso della RF per acquisire i numeri di serie richiede la manutenzione dell'infrastruttura RF (rete e hardware), oltre alla gestione del software (ITS Mobile o SAPConsole). Ciò aggiunge costi all'azienda sia per l'IT che per il magazzino. Se però all'interno del magazzino si fa già uso di questa tecnologia questo non comporta alcun costo aggiuntivo se non quello della progettazione di questa transazione.
- Catturare i numeri di serie consuma risorse di risorse di magazzino.
- Deve essere gestita la possibilità di lavorare alla scannerizzazione con più utenti alla volta.

Capitolo 3

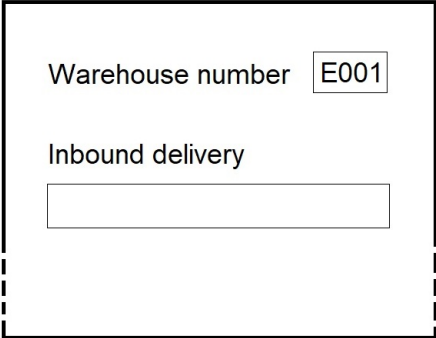
Progetto

In questo capitolo viene esposta la parte di progettazione ed implementazione della transazione in radio frequenza per la scannerizzazione dei numeri seriali dei prodotti in fase di ricezione merce. Nella prima parte, dopo aver presentato i requisiti, viene mostrata la personalizzazione delle impostazioni di sistema per poter codificare il dispositivo palmare utilizzato dagli operatori di magazzino. Nella seconda parte viene invece illustrata come è stata implementata la logica di funzionamento della transazione.

3.1 Requisiti

Si vuole gestire a livello di item della delivery il numero di serie dei prodotti. Per fare questo si necessita di una transazione in radio frequenza che catturi i seriali ottimizzando i tempi di inserimento rispetto a quelli manuali. Specificatamente occorre implementare una transazione RF che permetta di inserire all'interno di una inbound delivery ed essenzialmente all'interno del sistema che utilizza il modulo EWM i codici seriali dei prodotti in arrivo al magazzino relativi a quella specifica inbound delivery.

Sono necessarie due schermate per questa transazione:



Warehouse number

Inbound delivery

Figura 3.1: Mockup prima schermata

- La prima schermata deve permettere di inserire il numero di inbound delivery in EWM e in aggiunta la possibilità di visualizzare il numero di warehouse number in cui la risorsa è loggata (Fig. 3.1).

Warehouse number

Inbound delivery

SNCr /

SerialNo

Pro.
SNCr /

Pro.
SNCr /

Figura 3.2: Mockup seconda schermata

- La seconda schermata invece deve mostrare le informazioni della prima schermata ed in aggiunta alcuni dettagli dell'inbound delivery (Fig. 3.2). In particolare devono essere mostrati un contatore fisso che rappresenta il numero di prodotti gestiti a seriale presenti nell'inbound delivery con relativo contatore generale che tiene traccia del numero di seriali scannerizzati. Per ogni prodotto gestito a seriale inoltre deve essere presente una riga che ne mostri il nome e due contatori come i precedenti ma relativi solo allo specifico materiale. Infine deve essere possibile scannerizzare i seriali dei prodotti relativi a tale inbound delivery e alla pressione del tasto salva postare le modifiche al documento a sistema.

Per di più deve essere possibile lavorare anche con più operatori alla volta in maniera autonoma senza creare inconsistenze.

La transazione deve essere lanciata in un palmare che abbia le dimensioni 16x20.

3.2 Fasi preliminari di ogni progetto SAP

Come per ogni progetto, sia che si sviluppi su SAP che in qualsiasi altro ambiente, occorre organizzare tutto il codice secondo una struttura ben definita e di facile gestione. Innanzitutto dovrà quindi esser configurato un **package** di sviluppo specifico del progetto. Il package è un contenitore utilizzato all'interno del sistema per collegare gli oggetti di sviluppo - Function Modules, Function Groups, Screens - con gli strumenti di organizzazione propri di Application Server ABAP. Sono progettati quindi per aiutare gli sviluppatori a modulare, incapsulare e disaccoppiare le unità nel sistema SAP. In aiuto al programmatore, la transazione **Object Navigator SE80** permette di navigare tra i package definiti nel sistema selezionando lo strumento Repository Browser e modificare tutto il codice sviluppato come un vero e proprio editor (Fig. 3.3). In questo particolare progetto è stato creato quindi il package **ZSCWM_RF_INSN** seguendo la convenzione stabilita da SAP, cioè che tutti gli oggetti non definiti dallo standard devono cominciare con la lettera "Z" o "Y". Si potranno dunque creare tutti gli oggetti relativi al progetto mantenendo la modularità e il disaccoppiamento dagli altri oggetti del sistema.

Object Name	Description
▼ ZSCWM_RF_INSN	Package RF Inser Serial in Inbound Delivery
▼ Dictionary Objects	
▶ Table Types	
▶ Structures	
▼ Function Groups	
▼ ZSCWM_RF_INSN	F. group RF ins. serial inbound del.
▶ Function Modules	
▶ Dictionary Structures	
▶ Type Groups	
▶ Fields	
▶ Subroutines	
▶ PBO Modules	
▶ PAI Modules	
▶ Screens	
▶ Includes	

Figura 3.3: Struttura unità di progetto

Prima di iniziare lo sviluppo sono state create due **Change Request (CR)** per il trasporto delle modifiche dal sistema di sviluppo verso quello di testing in quality. Come descritto nel primo capitolo, anche in questo caso la piattaforma SAP è stata configurata scegliendo di suddividerla in tre istanze separate, rispettivamente Sviluppo, Controllo qualità e testing e Produzione. In una CR di tipo workbench sarà contenuto il codice applicativo, mentre nell'altra di tipo customizing saranno contenute tutte le modifiche eseguite dalla transazione SPRO.

3.3 Customizing

Oltre a tutti i programmi ABAP che implementano le logiche di processo devono essere definite le impostazioni e variabili d'ambiente che regolano i processi funzionali che avvengono nel sistema. Queste modifiche alle impostazioni costituiscono il **Customizing** di un singolo sistema e possono essere effettuate grazie alla transazione **SAP Project Reference Object SPR0**. Questa transazione presenta un menu ad albero suddiviso in cartelle, innestate su più livelli, in base al processo funzionale di riferimento. In questo caso il percorso di riferimento è: **SAP Reference IMG -> SCM ExtendedWarehouse Management -> Extended Warehouse Management -> Mobile Data Entry -> Radio Frequency (RF) Framework**

Come descritto nel primo capitolo occorre quindi definire nel customizing per ogni transazione RF che si voglia implementare: un display profile, un presentation profile, un personalization profile oltreché la transazione logica.

3.3.1 Display Profile

Definizione del **Display Profile** specifico per i requisiti che identifica le dimensioni dello schermo. La creazione del display profile viene fatta con le relative misure specifiche ad hoc dei dispositivi palmari in dotazione agli operatori di magazzino. Il nuovo display profile ha quindi le dimensioni dello schermo 16x20, a differenza di quello standard 8x40.

3.3.2 Presentation Profile e Personalization Profile

Il **Presentation Profile**, insieme al **Personalization Profile**, viene utilizzato per supportare diverse strutture di menu per utenti diversi. SAP infatti utilizza questi due profili per stabilire come il sistema deve comportarsi.

Una volta definito il presentation profile viene associato ad esso il personalization profile. Il presentation profile deve essere poi anche associato ad un intero magazzino attraverso la transazione: **Assign Presentation Profile to Warehouse**. Assegnando quindi al magazzino di riferimento il presentation profile. Al personalization profile vengono invece collegati i singoli utenti che utilizzano le risorse di sistema, permettendo di configurarne il menu di transazioni accessibili. Attraverso la transazione **/SCWM/RSRC** si possono creare le vere e proprie risorse che identificano gli operatori all'interno del magazzino. In questo caso è stata definita una risorsa appartenente al gruppo di risorse dedicate alle operazioni di entrata merci ed associate al presentaion device definito in precedenza.

3.3.3 Presentation Device

Il passo successivo è la creazione del **Presentation Device** a cui associare il display profile appena creato. Il presentation device raffigura quindi il collegamento tra la risorsa fisica utilizzata dagli operatori ed il display profile. In fase di Logon all'interno dell'ambiente RF l'operatore dovrà inserire il codice identificativo del proprio presentation device affinché il sistema possa recuperare il profilo associato alla risorsa e mostrare l'interfaccia utente adeguata. La gestione del presentation device del sistema non fa parte delle transazioni comprese nelle attività di Customizing accessibili mediante SPRO ma è necessario usare la transazione /SCWM/PRDVC. Questa altro non è che un'interfaccia per l'interazione con la tabella di sistema che mantiene i record di ogni presentation device.

3.3.4 Menu Manager

L'ultima fase di configurazione delle risorse è la definizione del menu di transazioni logiche in radiofrequenza mostrato all'utente dopo aver effettuato l'accesso all'ambiente di esecuzione RF environment. La possibilità di personalizzare il menu in base ai compiti di un operatore permette di stabilire con precisione quali sono le transazioni a cui l'utente può accedere, nascondendogli in questo modo le parti del sistema non di sua competenza.

Anche in questo caso il menu è stato creato in copia a partire da quello standard offerto da SAP, associando il nuovo elemento ai profili creati precedentemente. Una volta terminata la creazione, sono state eliminate le voci relative ai processi che non fanno parte dell'entrata merci inserendo in aggiunta la voce della nuova transazione logica ZINSN (Fig. 3.4).


Menu Hierarchy -Pres.Texts(Size:40)	Object	Sequence
▼ Main Menu	MAIN	
▼ System-Guided	SYGUID	01
▼ Manual Selection	MANSEL	02
▼ Inbound Processes	IBMAIN	03
▼ Unloading	ULMAIN	01
▼ Deconsolidation	SPMAIN	02
▼ Putaway	PTMAIN	03
▼ Receiving of Handling Units	RHMAIN	04
•  Insert Serial Number in ID	ZINSN	05
▼ Outbound Processes	OBMAIN	04
▼ Internal Processes	IMMAIN	05

Figura 3.4: Definizione delle voci del Menu Manager

3.4 Le interfacce

Come da requisiti è necessario creare due schermate. In particolare una prima schermata **001** che permetta all'operatore di inserire il numero di inbound delivery e una seconda schermata **002** che invece mostra le informazioni dei serial numbers per quella inbound delivery.

3.4.1 Scambio di informazioni tra interfacce

Sap mette a disposizione principalmente due modalità per far in modo che le schermate si scambino le informazioni tra uno step e l'altro (Fig. 3.5).

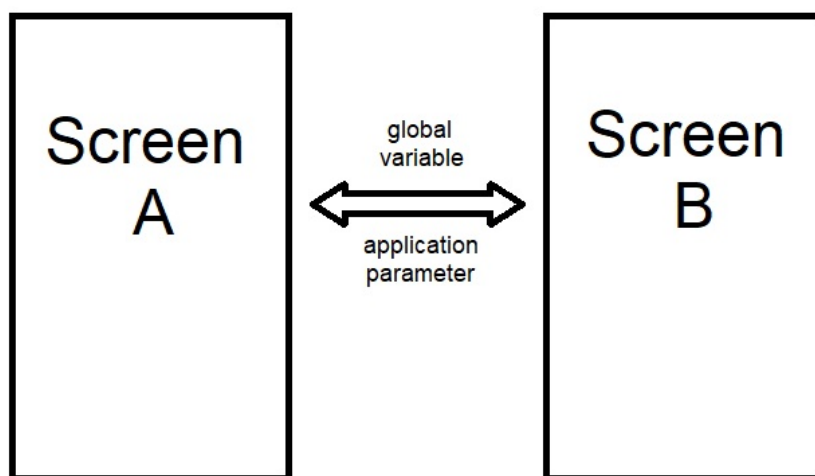


Figura 3.5: Scambio di informazioni tra schermate RF

- La prima meno consigliata è l'utilizzo di variabili globali.
- La seconda invece è quella standard e consiste nell'utilizzo degli **Application Parameter**. All'interno del package ed in particolare quindi all'interno del dizionario del sistema si possono definire strutture e tabelle custom che possono poi essere esplicitate nei application parameter. Questi quindi conterranno le informazioni che la transazione vuole elaborare e salvare, ma anche le informazioni che verranno mostrate o inserite nello schermo e che possono essere scambiate tra uno screen e l'altro.

3.4.2 Prima Schermata

La creazione di un nuovo Screen ABAP è possibile utilizzando la transazione SE80, corrispondente a Object Navigator. Successivamente grazie al programma **Screen Painter** è possibile modificarne il contenuto.

Il requisito richiedeva quindi una prima schermata che mostrasse all'operatore il numero di magazzino a cui ha effettuato l'accesso e la possibilità di inserire in un campo inputabile il numero dell'inbound delivery alla quale vuole poter assegnare i serial number (Fig. 3.6). Questi contenitori sono associati quindi alle strutture definite nel dizionario che a loro volta sono esplicitate come application parameter.

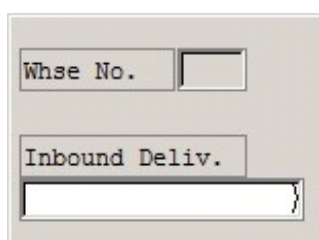


Figura 3.6: Screen Painter della schermata 001

3.4.3 Seconda schermata

Stesso ragionamento deve esser effettuato per la seconda schermata (Fig. 3.6).

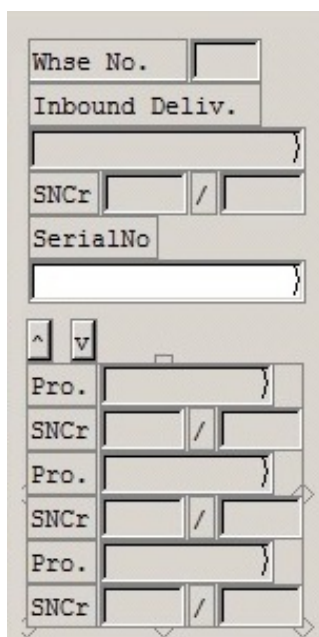


Figura 3.7: Screen Painter schermata 002

Questa schermata risulta più complessa già da requisito. In particolare essa dovrà mostrare all'operatore:

- Il numero di magazzino **Whse No.** a cui ha effettuato l'accesso;
- Il numero di inbound delivery **Inbound Deliv.** relativa all'arrivo della merce in magazzino della quale non è ancora stata effettuata la goods receipt e che ha alcuni materiali gestiti a seriale;
- Un contatore che tenga traccia dei seriali già scannerizzati / su il numero totale di prodotto da scannerizzare;
- Un campo inputabile in cui è possibile inserire il numero di seriale;
- Uno step loop nel quale per ogni riga venga mostrato il nome del materiale gestito a seriale **Pro.** che appartiene alla inbound delivery e il relativo conteggio parziale per quel determinato prodotto, dato dalla somma dei prodotti di quel tipo per ogni item.

3.5 ABAP Dictionary

Così come per il linguaggio SQL anche ABAP può essere diviso in due parti:

- DML (Data Manipulation Language)
- DDL (Data Definition Language)

Una parte quindi di comandi di query e aggiornamento dei dati (DML) ed una parte invece di definizione di dati (DDL).

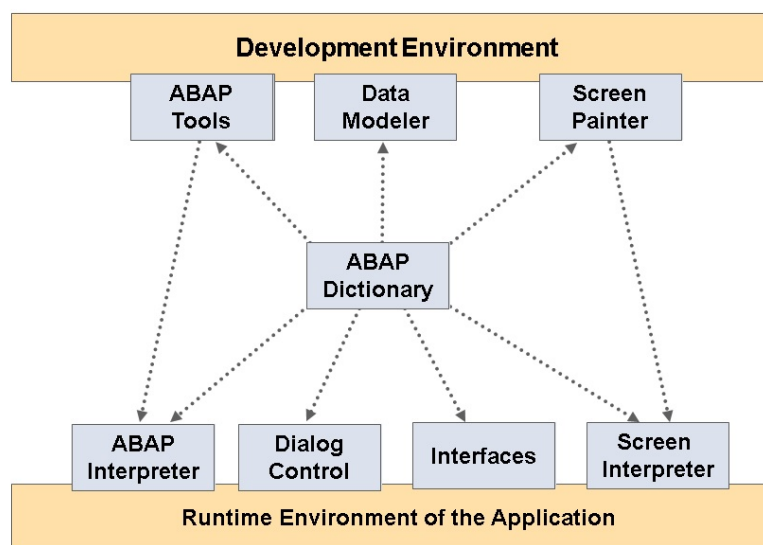


Figura 3.8: ABAP Dictionary

Si utilizza quindi il **ABAP Dictionary** per creare e gestire le definizioni dei dati (**metadati**). Il dizionario ABAP consente una descrizione centrale di tutti i dati utilizzati nel sistema senza ridondanze, le informazioni nuove o modificate vengono fornite automaticamente per tutti i componenti del sistema (Fig. 3.8). Ciò garantisce l'integrità, la coerenza e la sicurezza dei dati. Il dizionario ABAP supporta la definizione di tipi definiti dall'utente (come i tipi di strutture e di tabelle) permettendo quindi di creare gli oggetti corrispondenti nel database relazionale sottostante utilizzando queste definizioni di dati. Il dizionario ABAP descrive pertanto la struttura logica degli oggetti utilizzati nello sviluppo dell'applicazione e mostra come sono mappati al database relazionale sottostante. L'ABAP Dictionary è completamente integrato in ABAP Workbench. Il sistema SAP funziona in modo interpretativo, consentendo al dizionario ABAP di essere attivamente integrato nell'ambiente di sviluppo, in questo modo gli interpreti vedono solo le rappresentazioni interne degli oggetti. Queste rappresentazioni interne vengono regolate automaticamente quando il sistema rileva che sono state apportate modifiche nel dizionario ABAP. Ciò garantisce che gli interpreti degli screen e di ABAP, l'interfaccia di database e gli strumenti di sviluppo accedano sempre ai dati correnti.

3.5.1 Dictionary Object

All'interno della transazione occorre utilizzare delle strutture custom che memorizzano vari dati ed in particolare i dati relativi ai contenitori delle schermate che sono definite nell'ABAP Dictionary ed esplicitate negli Application Parameter. In particolare:

- **ZCS_SN_MM** che mantiene le informazioni generali relative all'inbound delivery (Fig. 3.9).

Component	Typing M...	Component Type	Data ...	L...	D...	Short Description
<u>DOCNO_ID</u>	1 Types	/SCWM/SP_DOCNO_PDI	CHAR	35	0	Inbound Delivery
<u>LGNUM</u>	1 Types	/SCWM/LGNUM	CHAR	4	0	Warehouse Number/Warehouse Complex
<u>SERID</u>	1 Types	/SCWM/DE_RF_SERID	CHAR	30	0	RF: Serial Number
<u>COLSN</u>	1 Types	/SCWM/DE_RF_COLSN	NUMC	5	0	RF: Number of Serial Numbers Created
<u>NOSN</u>	1 Types	/SCWM/DE_RF_NOSN	NUMC	5	0	RF: Number of Serial Numbers for Selection

Figura 3.9: Struttura ZCS_SN_MM

Essa è costituita da vari campi:

- **DOCNO_ID** è la struttura che rappresenta l'inbound delivery;

- **LGNUM** rappresenta invece il numero di magazzino;
 - **SERID** invece equivale al serial number inserito dall'utente quando scannerizza un prodotto;
 - **COLSN** e **NOSN** invece saranno i contatori rispettivamente dei seriali già scannerizzati e dei seriali totali dei prodotti.
- **ZCS_SN_MATERIAL_MM** che invece tiene traccia delle informazioni presenti in una riga della tabella (Fig. 3.10).

The screenshot shows the SAP Structure configuration for ZCS_SN_MATERIAL_MM. The structure is active and has the short description 'Structure for set loop in RF insert Serial'. It contains five predefined components:

Compon...	Typing M...	Component Type	Dat...	L...	D..	Short Description
<u>MATNR</u>	1 Types	/SCWM/DE_RF_MATNR	CHAR	40	0	Product Number for Entry
<u>COLSN</u>	1 Types	/SCWM/DE_RF_COLSN	NUMC	5	0	RF: Number of Serial Numbers Created
<u>NOSN</u>	1 Types	/SCWM/DE_RF_NOSN	NUMC	5	0	RF: Number of Serial Numbers for Selection
<u>PRODUCT</u>	1 Types	/SCDL/DL_PRODUCT_STR		0	0	Product
<u>SERIALN</u>	1 Types	/SCWM/DL_SN_DATA_TAB		0	0	Serial Numbers for Delivery Item

Figura 3.10: Struttura ZCS_SN_MATERIAL_MM

Dove:

- **MATNR** è il nome del materiale;
 - **COLSN** e **NOSN** sono rispettivamente i contatori dei seriali già scannerizzati e dei seriali totali per quello specifico tipo di materiale;
 - **PRODUCT** è una struttura che raccoglie tutte le informazioni di quel particolare materiale;
 - **SERIALN** è una tabella che invece memorizza tutti seriali scannerizzati per quello specifico materiale che ancora non sono stati assegnati alla inbound delivery.
- **ZCT_SN_MATERIAL_MM** è quindi la tabella che raccoglie tutte le righe dello step loop definite dalla struttura precedente necessario per memorizzare tutte le informazioni relative ad ogni tipo di materiale gestito a seriale presente nell'inbound delivery (Fig. 3.11).

The screenshot shows the SAP Table configuration for ZCT_SN_MATERIAL_MM. The table is active and has the short text 'Table for Material in steploop'. The 'Line Type' is set to 'ZCS_SN_MATERIAL_MM'.

Table Type	Table Name	Status
Table Type	ZCT_SN_MATERIAL_MM	Active
Short text	Table for Material in steploop	
Line Type	ZCS_SN_MATERIAL_MM	

Figura 3.11: Tabella ZCT_SN_MATERIAL_MM

3.6 La transazione ZINSN

La nuova transazione da implementare non può essere creata in copia a partire da un'altra dello standard in quanto il sistema non ha una transazione che svolga questo tipo di compito. Viene dunque definita una nuova entry nella tabella delle transazioni del sistema denominata **ZINSN** nella transazione Define Logical Transactions.

3.6.1 RF Framework Interaction Model

Come descritto nel primo capitolo, l'architettura del framework RF consiste in tre livelli che vengono innescati ogni qualvolta l'utente esegue un'azione sullo schermo.

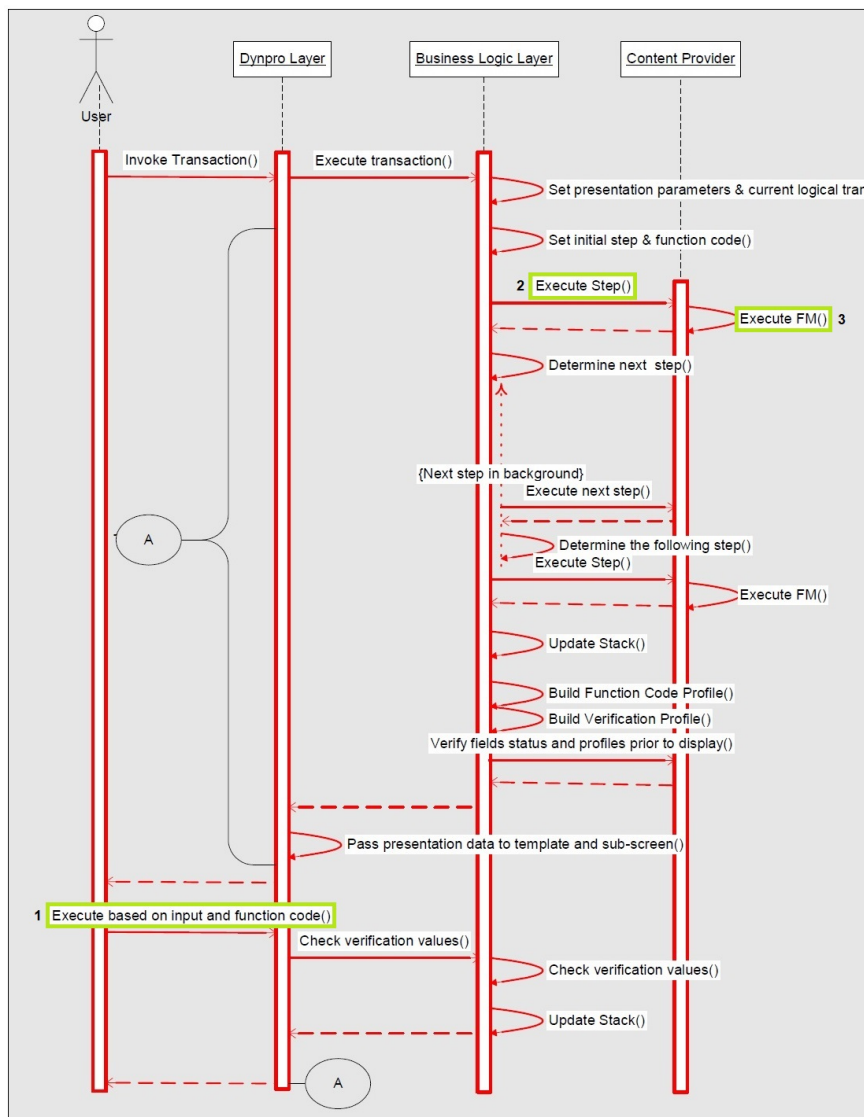


Figura 3.12: Modello di interazione del Framework RF

Occorre dunque definire quelli che sono gli **step** (punto 2 fig.3.12) della transazione in modo tale che il business logic layer possa eseguirlo e riconoscere quello che sarà lo step successivo. Gli step possono esser visti come degli stati in cui la transazione si trova e corrispondono ad una schermata. Inoltre occorre esplicitare quelli che sono i **function code** (punto 1 fig. 3.12). Essi corrispondono ad un pulsante nello schermo o semplicemente ad un'azione dell'utente come può essere la scannerizzazione di un codice a barre. Infine ogni qual volta si voglia elaborare dei dati bisogna eseguire dei **function module** processati dal content provider (punto 3 fig. 3.12).

3.6.2 Steps

Una volta definiti dall'analisi quelli che sono gli step e le relative schermate, essi devono esser definiti all'interno del sistema nella transazione **Define Steps**. In particolare vengono quindi specificati i due step della transazione (Fig. 3.13):

- **ZIDSEL** che rappresenta il primo passo della transazione dove occorre inserire il numero di inbound delivery;
- **ZISNID** che invece rappresenta il secondo passo dove vengono inseriti i serial numbers dei prodotti relativi all'inbound delivery.

Define Steps	
Step	Description
ZAHPRI	Z: Adhoc Pro. WT - Main screen
ZIDSEL	Inbound Delivery Selection
ZINDLV	Z: Scan Inbound Delivery Number
ZINSSN	Serial Number Packing
ZISNID	Insert SN in Material of Inb. Del.

Figura 3.13: Definizione degli step della transazione ZINSN

3.6.3 Function Code Profile

Per ogni step è possibile definire alcuni tasti specifici per quella schermata. In particolare, da requisiti, per ogni schermata sarà necessario poter tornare indietro **BACKF**, ripulire i campi di input **CLEAR**, dare la possibilità di inserimento **ENT** che nella seconda schermata corrisponde anche ad una scannerizzazione del seriale e sempre in quest'ultima schermata la possibilità di poter salvare **SAVE**. Per fare questo è necessario definire nella **Define Function code Profile** i function code per ogni step.

3.6.4 Architettura

La transazione si muove su due step principali, ognuno collegato ad una schermata (Fig. 3.14). Si avrà quindi uno step iniziale ZIDSEL collegato alla prima schermata di inserimento della inbound delivery ed un altro step ZISNID collegato alla seconda schermata nella quale sono mostrate le informazioni della delivery e la possibilità di modificare quest'ultima scannerizzando i serial number della merce.

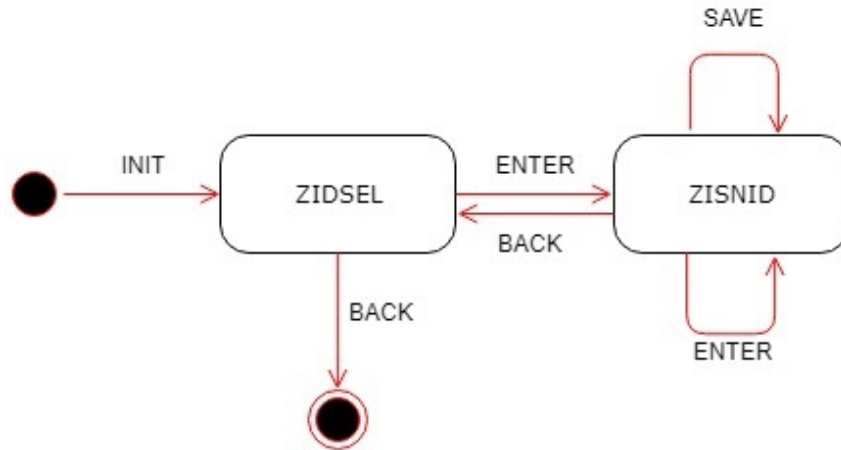


Figura 3.14: Diagramma degli stati della transazione ZINSN

All'interno della schermata relativa alla transazione ZINSN di Customizing è presente la sezione denominata Define Logical Transaction Step Flow. Qui è possibile definire, attraverso ogni tupla della tabella (Fig. 3.15), una transizione da uno step ad un altro mediante l'esecuzione di un function module, invocato all'attivazione di un function code. Per ogni transizione vengono quindi dichiarati lo step successivo ed il tipo di processamento desiderato. Questi passaggi anche intra-step possono essere eseguiti in background oppure in foreground (ogni passaggio in foreground ha uno screen corrispondente che viene aggiornato).

Define Logical Transaction step flow									
Pres.P.	Log.Tr...	Step	Func.C	Valid.Prof	Function Module	Next Step	Proc.Mode	BckgrfCode	
Z001	ZINSN	ZIDSEL	BACKF	<input type="checkbox"/>	ZSCWM_RF_ISN_SN_IN_BACK		1 Background	▼	CMPTRS
Z001	ZINSN	ZIDSEL	ENTER	<input type="checkbox"/>	ZSCWM_RF_ISN_SEL_ID_PAI	ZISNID	1 Background	▼	PB01
Z001	ZINSN	ZIDSEL	INIT	<input type="checkbox"/>	ZSCWM_RF_ISN_SEL_ID_INIT	ZIDSEL	1 Background	▼	PB01
Z001	ZINSN	ZIDSEL	PBO1	<input type="checkbox"/>	ZSCWM_RF_ISN_SEL_ID_PBO	ZIDSEL	2 Foreground	▼	
Z001	ZINSN	ZISNID	BACKF	<input type="checkbox"/>	ZSCWM_RF_ISN_SN_IN_BACK	ZIDSEL	1 Background	▼	PB01
Z001	ZINSN	ZISNID	ENTER	<input type="checkbox"/>	ZSCWM_RF_ISN_SN_IN_PAI	ZISNID	2 Foreground	▼	
Z001	ZINSN	ZISNID	PBO1	<input type="checkbox"/>	ZSCWM_RF_ISN_SN_IN_PBO	ZISNID	2 Foreground	▼	
Z001	ZINSN	ZISNID	PBO2	<input type="checkbox"/>	ZSCWM_RF_ISN_SN_IN_PBO_SAVE	ZISNID	2 Foreground	▼	
Z001	ZINSN	ZISNID	SAVE	<input type="checkbox"/>	ZSCWM_RF_ISN_SN_IN_PAI_SAVE	ZISNID	1 Background	▼	PB02

Figura 3.15: Step Flow di ZINSN

A fronte di questo per ogni interazione è stato quindi sviluppato un function module. I function module sono principalmente di due tipologie: **PBO (Process Before Output)** e **PAI (Process After Input)**. I function module di tipo PBO hanno l'obiettivo di recuperare ed elaborare le informazioni recuperate dal database prima di mostrarle all'utente andando a popolare la schermata. I secondi, invece, sono invocati ogni volta che viene recepito un input dall'ambiente di esecuzione. È in aggiunta necessario distinguere, in caso di un'applicazione che utilizzi i Dialog come una transazione RF, tra moduli PBO/PAI collegati ad un Function Module o a una schermata. La figura 3.16 mostra il flusso e l'interazione fra queste due categorie.

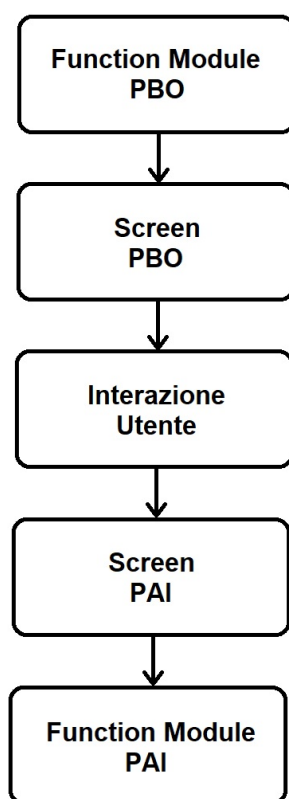


Figura 3.16: Flusso PAI/PBO

3.6.5 Function Modules

ZSCWM_RF_ISN_SEL_ID_INIT

Il primo passo da eseguire inizialmente corrisponde sempre allo step iniziale definito al momento della definizione della transazione logica ZINSN, dunque lo step ZIDSEL ed in particolare quello con function code INIT. Il function module ZSCWM_RF_ISN_SEL_ID_INIT avrà il compito di inizializzare i data container della transazione RF. Infatti prima di poter utilizzare i data container occorre informare il framework RF attraverso delle istruzioni. La registrazione dei data container è un'operazione obbligatoria da eseguire. Il framework fornisce i seguenti metodi:

- Il primo consiste nell'inizializzare i parametri:

Listing 3.1: Inizializzazione parametri

```
1 CALL METHOD /scwm/cl_rf_bll_srvc=>init_screen_param.
```

- Il secondo invece registra il parametro all'interno del framework RF. Per esempio:

Listing 3.2: Settaggio parametri

```
1 CALL METHOD /scwm/cl_rf_bll_srvc=>set_screen_param
2   EXPORTING
3     iv_param_name = 'CS_SN_MM'.
```

I function module possono utilizzare un qualsiasi numero di tabelle, ma ogni schermata può presentare sì più strutture ma solamente una tabella. Occorre quindi specificare quale tabella verrà presentata utilizzando il seguente metodo:

Listing 3.3: Settaggio tabella

```
1 CALL METHOD /scwm/cl_rf_bll_srvc=>set_scr_tabname
2   EXPORTING
3     iv_scr_tabname = 'ZCT_SN_MATERIAL_MM'.
```

ZSCWM_RF_ISN_SEL_ID_PBO

Il passo successivo dunque sarà relativo all'esecuzione del function module associato all'ultimo function code che era stato lanciato dallo step precedente, cioè quello del PBO1. Questo fm di tipo PBO andrà a ricavare e riempire la prima schermata con le informazioni richieste da requisito, cioè il numero di magazzino con qui si è loggata la risorsa.

ZSCWM_RF_ISN_SEL_ID_PAI

Il prossimo cambiamento sarà dovuto quindi dall'interazione con l'utente. Nel caso quest'ultimo decidesse di inserire il numero di inbound delivery all'interno del campo, alla pressione del tasto ENTER verrà eseguito il function module di PAI. A questo punto il fm controlla il corretto numero di inbound delivery e in tal caso si salverà a sistema tutte le informazioni necessarie per lo step successivo.

Per selezionare l'inbound delivery SAP fornisce il metodo **QUERY** al quale posso dare in ingresso complessi criteri di selezione. In particolare:

- ID number dato in ingresso dall'utente
- Warehouse Number

Inoltre è possibile ricavare solo le informazioni di interesse attraverso una `read_option`, in questo caso per esempio si vorrà poter sapere lo stato degli item e i relativi serial number. Nel caso non si riuscisse a ricavare l'inbound delivery perché non esiste o non è assegnata a quel warehouse number verrà catturata l'eccezione e verrà lanciato un messaggio d'errore avvisando quindi l'utente del errato numero di inbound delivery in modo tale che potrà riprovare con quello corretto. Se l'inbound invece è corretta viene salvato a livello globale le varie informazioni della delivery. A questo punto prima di procedere, con il function module successivo che andrà a riempire tutti i campi della schermata aggiornando lo steploop mostrando come prima riga l'ultimo prodotto scannerizzato, occorre verificare varie condizioni:

- se l'inbound delivery non è già stata postata con goods receipt (in questo caso verrà visualizzato un messaggio di errore e l'utente potrà provare con un'altra);
- se ha materiali all'interno dell'inbound gestiti a seriale (se non ha nessun materiale gestito a seriale anche in questo caso verrà mostrato un'errore e l'utente potrà provare con un'altra).
- se ha alcuni items gestiti a seriale già postati a goods receipt (in questo caso deve esser mostrato un pop-up di warning che avvisa l'utente della lista di materiali già postati).

ZSCWM_RF_ISN_SN_IN_BACK

Nel caso l'utente decidesse di tornare indietro e uscire dalla transazione occorre settare il function code di uscita dalla transazione CMPTRS. Inoltre all'interno del function module sarà necessario ripulire tutte le variabili, strutture, tabelle e oggetti della transazione. Questo function module è valido anche nel caso l'utente decidesse di tornare indietro dalla seconda schermata alla prima, in quanto anche in questo caso andranno ripulite tutte le variabili.

ZSCWM_RF_ISN_SN_IN_PBO

Una volta ricavate tutte le informazioni della inbound delivery questo function module dovrà settare tutti i campi della schermata con i dati ricavati. Oltre a settare il contatore generale dovrà esserci per ogni materiale gestito a seriale la descrizione con nome del materiale, il contatore dei seriali scannerizzati e il contatore totale di prodotti di quel materiale nell'inbound delivery. Per fare questo occorre scorrere ogni item della inbound delivery e dopo aver controllato lo stato e se gestito a seriale, raccogliere le informazioni ed aggiungerle nell'entry riferita a quel tipo di materiale della tabella dello step loop.

ZSCWM_RF_ISN_SN_IN_PAI

Al momento della scannerizzazione di un serial number si eseguirà il function module ZSCWM_RF_ISN_SN_IN_PAI. Dopo una serie di controlli come:

- Seriale del formato corretto;
- Seriale corrispondente ad un materiale che sia all'interno di quell'inbound delivery;
- Seriale non sia già a sistema;
- Seriale non sia già stato scansionato anche se ancora non salvato a sistema;
- Il numero di seriali per quel tipo di materiale siano già stati letti tutti quanti;

Si potrà salvare quel seriale nella tabella dei materiali ZCT_SN_MATERIAL_MM relativo all'entry corretta e si aggiorneranno i contatori relativo al prodotto e quello generale.

ZSCWM_RF_ISN_SN_IN_PAI_SAVE

Questo è un punto cruciale della transazione. Il function module deve creare i seriali e inserirli negli item corretti dei prodotti scannerizzati fino a quel momento. Fondamentale inoltre è riuscire a salvare i seriali scannerizzati che però, nel frattempo, non siano già stati inseriti nell'inbound delivery da un altro utente. Per fare questo occorre rileggere l'inbound delivery per avere le informazioni più recenti: andare in **lock** sul documento in modo tale che nessun altro possa in quel momento modificare l'inbound delivery e spalmare tutti i seriali scannerizzati negli item corretti. Occorre poi salvare i cambiamenti effettuati all'inbound, slockare il documento.

ZSCWM_RF_ISN_SN_IN_PBO_SAVE

Dopo aver salvato l'inbound delivery occorre semplicemente richiamare i due function module usati precedentemente ZSCWM_RF_ISN_SEL_ID_PAI e ZSCWM_RF_ISN_SN_IN_PBO per resettare e visualizzare la schermata dell'inbound con i valori aggiornati.

3.7 Lock Object

Molto importante per avere l'accesso sincronizzato ai dati come in questo caso per le modifiche ai documenti di inbound delivery, è l'utilizzo della funzionalità **Lock Object**. Questa è infatti una funzionalità offerta da ABAP Dictionary che viene utilizzata per sincronizzare l'accesso ai dati da più di un programma. I lock object vengono utilizzati in SAP per evitare incoerenza quando i dati vengono inseriti o modificati nel database.

ABAP mette a disposizione il metodo lock, nello specifico per mantenere l'esclusivo accesso al documento di inbound delivery:

Listing 3.4: Metodo ABAP di Lock

```
1 lo_sp→lock(  
2   EXPORTING  
3     inkeys = lt_sp_k_head "Inbound Delivery"  
4     aspect = /scdl/if_sp_c⇒sc_asp_head  
5     lockmode = /scdl/if_sp1_locking⇒sc_exclusive_lock  
6   IMPORTING  
7     rejected = lv_rejected  
8     return_codes = lt_return_codes ).
```

3.8 Test

Tutti i test sono stati effettuati su un database reale. In particolare quindi il funzionamento corretto si basa effettivamente su dati e numeri di serie reali del magazzino di riferimento, mentre gli errori che sono stati gestiti sono dovuti appunto all'inesattezza rispetto al database ed ai requisiti di progetto.

3.8.1 Fasi preliminari: Creazione di una Inbound Delivery

Attraverso la transazione MIGO in SAP ERP è possibile effettuare un'entrata merci attraverso il tipo movimento **561** che crea a sistema una **Inbound delivery** (Fig. 3.17).

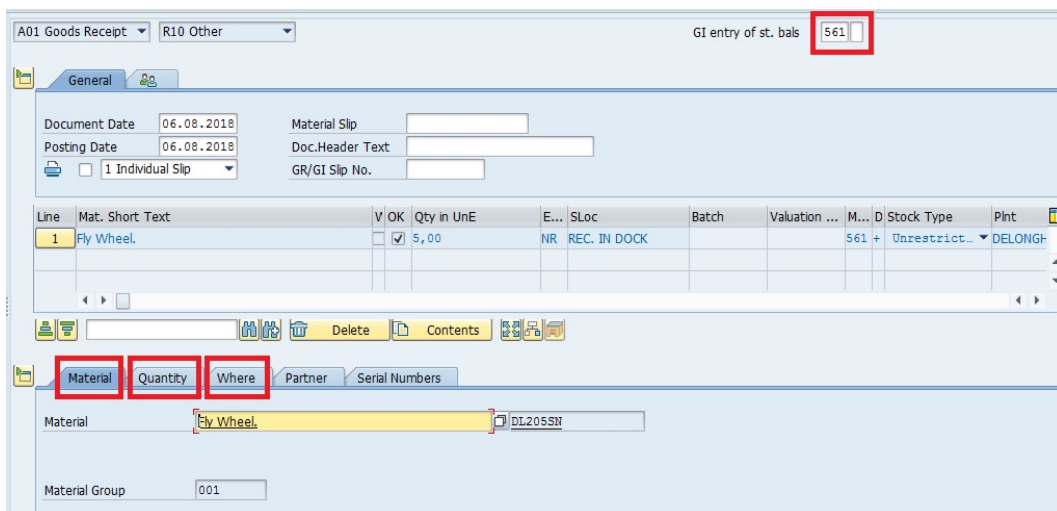


Figura 3.17: Transazione MIGO

In particolare si inseriranno i vari materiali (**Material**) con le relative quantità (**Quantity**) nel magazzino di riferimento (**Where**) identificato dall'accoppiata Plant e Storage Location, per ogni Item dell'inbound delivery. Una volta compilato il documento, viene postato e collegato ad un identificativo sul sistema ERP che a sua volta è collegato con un altro identificativo sul sistema EWM. Ora dunque sarà possibile visualizzare e modificare il documento di inbound delivery sul sistema EWM attraverso la transazione /SCWM/PRDI. Inserendo l'ID dell'inbound delivery creata su sistema ERP si potrà visualizzare e modificare il documento in EWM. È in questa transazione che un utente dovrebbe inserire manualmente i serial number per ogni prodotto gestito a seriali di ogni Item per poter poi eseguire la goods receipt (Fig. 3.18).

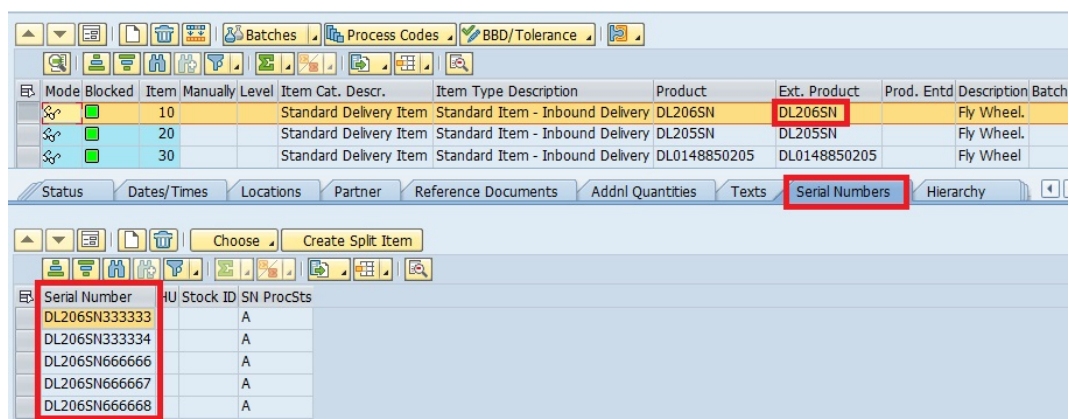


Figura 3.18: Serial Numbers nella Inbound Delivery, transazione /SCWM/-PRDI

3.8.2 Dimostrazione dell'uso e del corretto funzionamento della transazione singolo utente

Come inbound delivery di riferimento per questo test è stata utilizzata la numero **6103**, con le seguenti informazioni di principale rilevanza (Tab. 3.1 e Tab. 3.2):

ITEM	PRODUCT	SN	QTY	ALREADY SN	MISSING SN
10	DL206SN	C	5	1	4
20	DL205SN	C	2	1	1
30	DL0148850205		15	0	0
40	DL333SN	C	5	2	3
60	DL444SN	C	1	1	0
70	DL777SN	C	5	0	5
80	DL888SN	C	6	0	6
90	DL0148850205		10	0	0
100	DL999SN	C	1	1	0
110	DL205SN	C	5	0	5
120	DL555SN	C	10	6	4
130	DL444SN	C	1	1	0
140	DL444SN	C	3	0	3
160	DL888SN	C	5	0	5

Tabella 3.1: Tabella informazioni iniziali Inbound delivery

PRODUCT	SN	QTY	ALREADY SN	MISSING SN
DL206SN	C	5	1	4
DL205SN	C	7	1	6
DL333SN	C	5	2	3
DL444SN	C	5	2	3
DL777SN	C	5	0	5
DL888SN	C	11	0	11
DL999SN	C	1	1	0
DL555SN	C	10	6	4

Tabella 3.2: Tabella informazioni iniziali dei seriali Inbound delivery

TOTALE 13/49

L'utente in fase di logon dovrà accedere al numero di magazzino **E001** con la risorsa **INSNM** e presentation Device **Z001**. Selezionare le transazioni relative all'inbound process e infine selezionare la transazione creata precedentemente **Insert SN in ID** (Fig. 3.19).

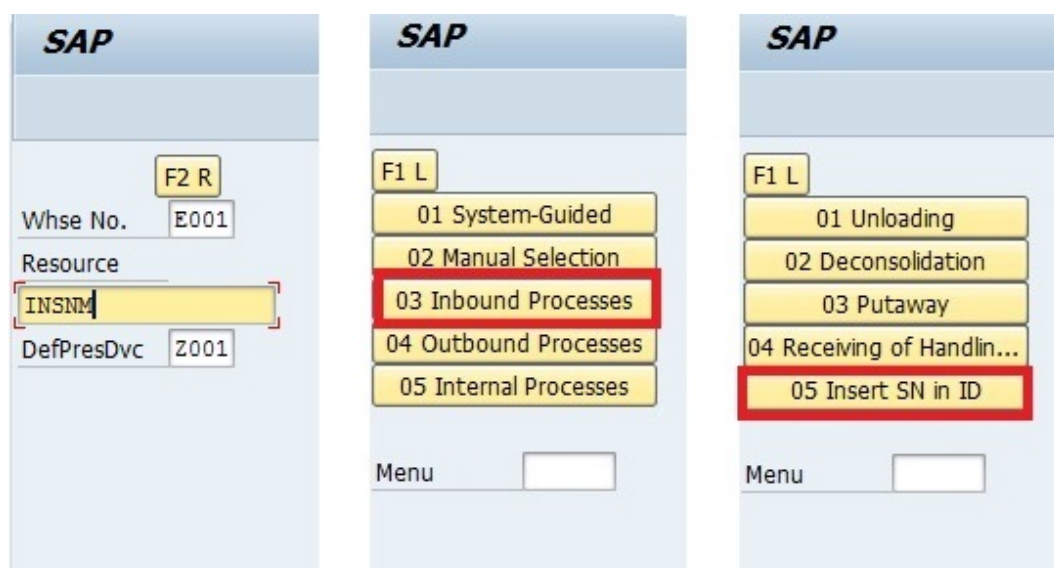
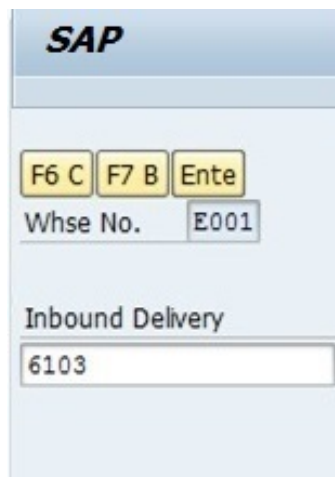


Figura 3.19: Percorso Menu transazione ZINSN

All'utente verrà mostrata quindi la prima schermata della transazione, dove dovrà inserire il numero di Inbound delivery. (Fig. 3.20)



SAP

F6 C F7 B Ente

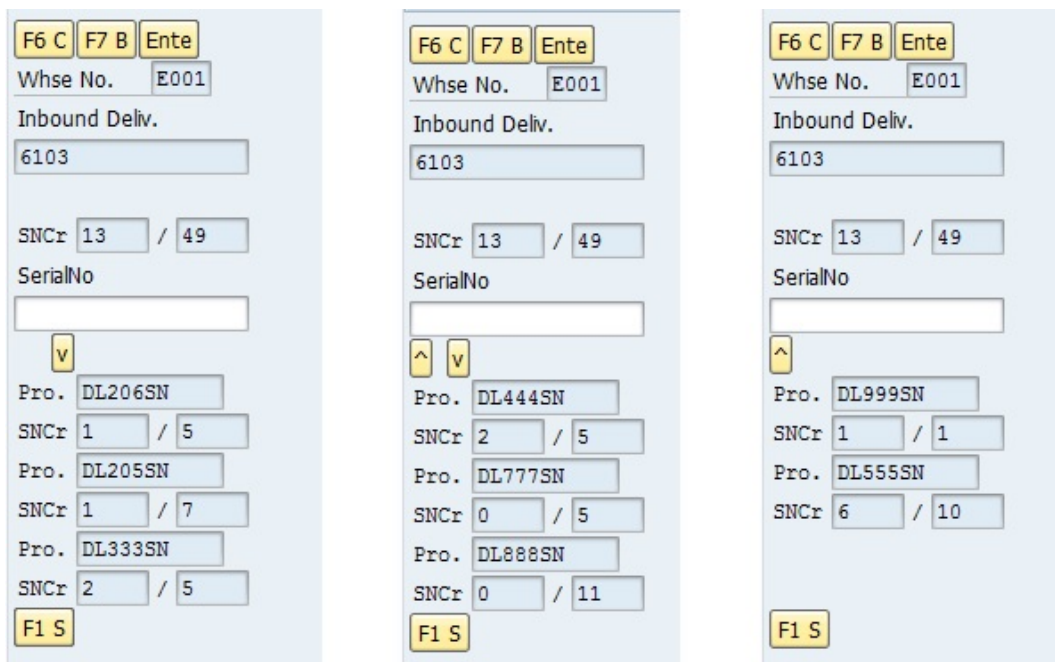
Whse No. E001

Inbound Delivery

6103

Figura 3.20: Prima Schermata: inserimento Inbound Delivery

Una volta inserito il numero di inbound delivery 6103 verrà mostrata la seconda schermata dove si avranno i contatori generali e per ogni prodotto esattamente come ci si aspettava. I materiali non gestiti a seriale non compaiono (Fig. 3.21).



The figure shows three sequential SAP screens for entering serial numbers. Each screen displays the same header information: Whse No. E001 and Inbound Deliv. 6103. The first screen shows a list of products with their respective serial number counters (SNCr) and a 'v' button. The second screen shows the same list with '^' and 'v' buttons. The third screen shows the same list with '^' buttons and an 'F1 S' button.

Product	SNCr	Total
DL206SN	1	5
DL205SN	1	7
DL333SN	2	5
DL444SN	2	5
DL777SN	0	5
DL888SN	0	11
DL999SN	1	1
DL555SN	6	10

Figura 3.21: Seconda Schermata: inserimento Serial Numbers

Ora si inseriscono un po' di seriali corretti per i relativi prodotti. In figura 3.22 è mostrato l'inserimento del primo numero seriale DL333SN777414

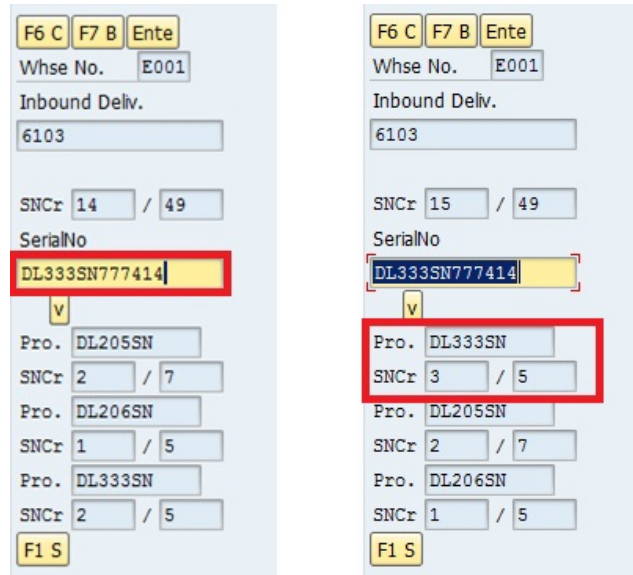


Figura 3.22: Inserimento Seriale DL333SN777414

Per un inserimento totale di **13 seriali** diversi rispetto l'inbound delivery iniziale. L'utente ora salva premendo il tasto **F1 Save**, torna indietro premendo **F7 Back** per accertarsi che l'inbound delivery sia stata davvero modificata dopo il Save. Reinserisce nella prima schermata il numero di inbound 6103, e può quindi verificare che quest'ultima è stata correttamente modificata e salvata a sistema (Fig. 3.23).

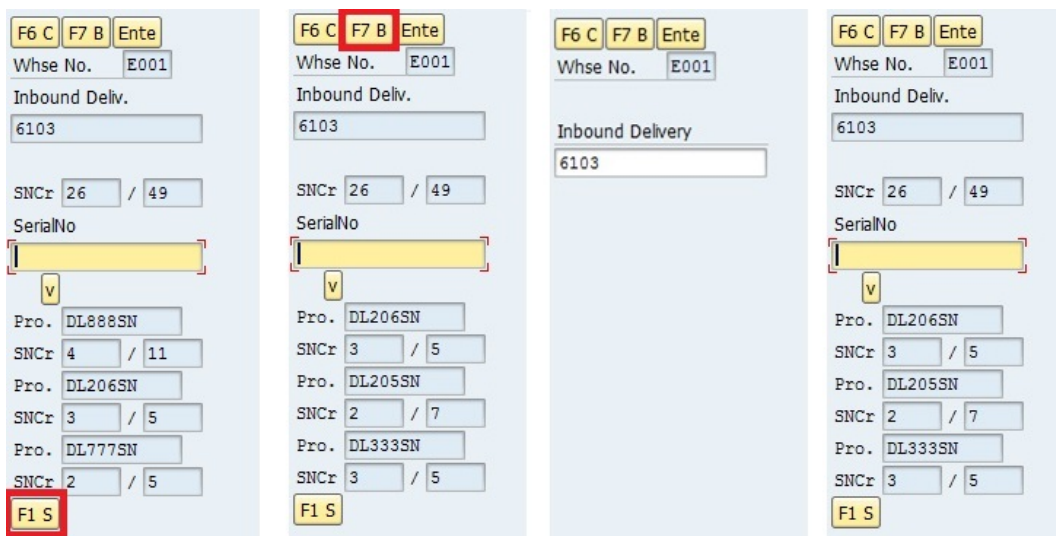


Figura 3.23: Save, Back e reinserimento Inbound Delivery 6103

3.8.3 Dimostrazione dell'uso e del corretto funzionamento della transazione più utenti alla volta

Un **utente A** accede alla transazione inserendo il numero di Inbound Delivery corrispettiva all'arrivo della merce in magazzino. Stessa cosa effettua l'**utente B**. Entrambi quindi inizialmente avranno la stessa schermata (contatore generale 26/49). L'utente A inserisce i seguenti SN: DL205SN123459, DL205SN123460, DL205SN123461 e DL888SN123461. Mentre l'utente B inserisce questi altri SN di cui il primo è lo stesso dell'utente A: DL205SN123459, DL205SN123470, DL777SN123488 (Fig. 3.24).

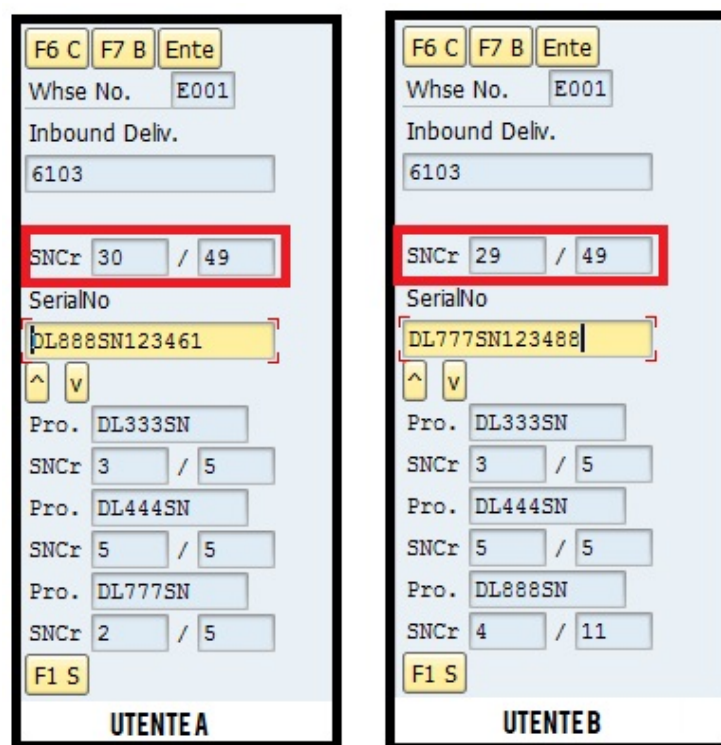


Figura 3.24: Schermata inserimento seriali: Utente A e Utente B

L'utente A decide di salvare per primo e inserire quindi tutti i seriali scannerizzati fino a quel momento all'interno della inbound delivery. Subito dopo l'utente B decide anche lui di salvare premendo il tasto F1 Save (figura 3.25). Giustamente l'utente B dopo il salvataggio non vede più solo 29 prodotti serializzati ma ben 32. Il contatore è impostato a 32 perché è dato dai 29 a cui era arrivato lui a scannerizzare più i 3 che ha aggiunto l'utente A. L'utente A ne ha aggiunti in verità 4 all'inbound delivery ma un SN era identico ad un seriale letto da B, quindi B avrà inserito effettivamente nell'inbound solo 2 dei 3 seriali scannerizzati perché uno era già stato salvato dall'utente A.

F6 C F7 B Ente
Whse No. E001
Inbound Deliv.
6103
SNCr 32 / 49
SerialNo
DL777SN123488
^ v
Pro. DL444SN
SNCr 5 / 5
Pro. DL777SN
SNCr 3 / 5
Pro. DL888SN
SNCr 5 / 11
F1 S
UTENTE B

Figura 3.25: Schermata Utente B dopo il salvataggio

L'utente A se fa un ulteriore salvataggio (anche senza inserire nessun altro seriale), oppure torna indietro F7 e rientra nell'inbound delivery vedendo i contatori aggiornati anche dai seriali dell'utente B (Fig. 3.26).

F6 C F7 B Ente
Whse No. E001
Inbound Deliv.
6103
SNCr 32 / 49
SerialNo
DL888SN123461
v
Pro. DL206SN
SNCr 3 / 5
Pro. DL205SN
SNCr 6 / 7
Pro. DL333SN
SNCr 3 / 5
F1 S
UTENTE A

Figura 3.26: Schermata Utente A dopo il secondo salvataggio

3.8.4 Gestione degli Errori e Warning

Sono stati gestiti tutti i possibili errori dove comparirà un messaggio di errore che avvisa l'utente e potrà quindi riprovare a inserire il parametro corretto:

- L'utente inserisce un numero di inbound delivery non esistente a sistema.

- L'utente inserisce un numero di inbound delivery già postata a goods receipt.
- L'utente inserisce un numero di inbound delivery con solo materiali non gestiti a seriale.
- L'utente scannerizza o inserisce un serial number del formato errato.
- L'utente scannerizza o inserisce un serial number già inserito.
- L'utente scannerizza o inserisce un serial number già salvato a sistema.
- L'utente scannerizza o inserisce un serial number corretto ma di un prodotto non presente nell'inbound delivery.
- L'utente scannerizza o inserisce un serial number di un prodotto per il quale sono già stati inseriti tutti i seriali.

È stato gestito anche il caso di un warning a cui l'operatore può essere interessato:

- L'utente inserisce un numero di inbound delivery con alcuni item già postati a goods receipt (dunque già serializzati) e altri ancora da serializzare. Comparirà quindi un popup di warning che avvisa l'utente della lista di materiali con goods receipt.

Conclusioni

L'obiettivo di questa tesi era dunque di realizzare una transazione in radiofrequenza non presente nello standard con la quale si possa inserire a sistema i numeri seriali dei prodotti che arrivano in magazzino in maniera velocizzata e con minor possibilità di errore. Dopo uno studio approfondito della piattaforma di riferimento SAP e il relativo modulo EWM, si sono analizzati i requisiti ed i problemi del progetto per arrivare poi ad una implementazione valida. Alla fine dello sviluppo della soluzione tutte le modifiche devono essere trasportate prima su sistema di test per essere maggiormente collaudate e una volta confermato il comportamento secondo le specifiche devono essere trasportate nel sistema di produzione. La nuova transazione verrà quindi aggiunta ai processi operativi già esistenti così da poter essere utilizzato dagli operatori del magazzino. In questo caso, lo sviluppo della transazione logica in radiofrequenza analizzato in questo elaborato è stato limitato all'implementazione nel sistema di sviluppo. L'incarico di trasportare le modifiche nei successivi sistemi e di effettuare i test sarà poi affidato ad un'altra figura aziendale. I test svolti già in fase di sviluppo hanno però confermato le previsioni, cioè una notevole diminuzione dei tempi di inserimento a sistema oltreché una notevole diminuzione degli errori nell'inserimento dei numeri seriali grazie alla scansione del codice a barre riferito al numero di serie. Questo è dovuto anche ad un'interfaccia grafica molto semplice e intuitiva, oltre alla comodità di utilizzare un palmare portatile dalle dimensioni ridotte rispetto un PC fisso.

In un futuro non molto lontano si prevede l'utilizzo di una tecnologia RFID che potrà ridurre ulteriormente i tempi di lettura dei numeri seriali rispetto alla scansione dei codici a barre. Questa tecnologia è basata sulla capacità di memorizzazione di dati da parte di particolari etichette elettroniche e sulla capacità di queste di rispondere all'interrogazione a distanza da parte di appositi apparati fissi o portatili. Questo consente una rilevazione automatica, massiva ed a distanza di oggetti sia statici che in movimento.

Per quanto concerne il periodo di tirocinio svolto presso l'azienda Engineering Ingegneria Informatica S.p.a è stato necessario un ampio lasso di tempo per lo studio dei manuali della piattaforma e del linguaggio di programmazione prima di poter cominciare ad implementare la soluzione. L'ambiente è fortemente strutturato e richiede allo sviluppatore l'apprendimento di regole di programmazione e di strumenti creati ad hoc per la gestione delle numerose funzionalità messe a disposizione. Durante questo periodo di apprendimento

autonomo sono stati anche svolti diversi corsi tenuti da altri dipendenti dell'azienda molto utili sia per il progetto sia per capire le tecnologie e soluzioni in questo ambito.

Ringraziamenti

Innanzitutto ringrazio il professor Matteo Golfarelli, relatore di questa tesi di laurea, oltre che per l'aiuto e la grande conoscenza che mi ha donato in questi anni, per la disponibilità e precisione dimostratemi durante tutto il periodo di stesura.

Un sentito ringraziamento va a tutti i dipendenti di Engineering Ingegneria Informatica s.p.a. presso la sede operativa di Rimini, che mi hanno dato l'opportunità di conoscere la piattaforma software SAP ERP ed avermi guidato nel mio percorso progettuale, dandomi la possibilità di proseguire lavorando con loro.

Infine, ultimo ma non per importanza, un ringraziamento speciale a mia madre e mio padre che, con il loro sostegno, sia morale che economico, mi hanno permesso di arrivare fin qui, contribuendo alla mia formazione personale.

Bibliografia

- [1] SAP Development. *RF Cookbook*. <https://archive.sap.com/documents/docs/DOC-63848>.
- [2] Sarah Frazier. *Complete ABAP*. 2017.
- [3] SAP. *SAP Help Portal*. <https://help.sap.com/viewer/index>.
- [4] Meagan White. *SAP EWM Architecture and Programming*. 2016.