



**Università degli Studi di Pisa  
Facoltà di Ingegneria**

**C.d.L. Magistrale in Ingegneria Informatica per la Gestione  
d'Azienda**

---

**Tesi di Laurea in  
Sistemi Informativi L-S**

**Sviluppo di un sistema di business  
intelligence utilizzando un approccio  
metodologico Lean-Agile**

Candidato  
**Vittorio Netti**

Relatore  
**Prof. Francesco Marcelloni**

Secondo Relatore  
**Prof.ssa Beatrice Lazzerini**

Relatore Aziendale  
**Dott. Paolo Rodeghiero**

## INDICE

Introduzione.....	i
Capitolo 1. I sistemi di data warehousing.....	1
1.1 Il Data Warehouse .....	1
L'architettura di un Data Warehouse.....	2
1.1.1 Il Data Mart .....	2
1.1.2 I metadati .....	3
1.1.3 Accesso ai dati: tipologie di applicazioni.....	3
1.2 Data Warehouse vs. Database OLTP.....	4
1.3 Il ciclo di vita di un Data Warehouse. ....	6
1.3.1 Pianificazione del Progetto.....	6
1.3.2 Definizione dei requisiti del Business .....	7
1.3.3 Il modello multidimensionale.....	7
1.3.4 Il progetto fisico .....	7
1.3.5 Extraction, Transformation & Loading (ETL) .....	7
1.3.6 Il progetto dell'architettura tecnologica .....	8
1.3.7 Selezione dei prodotti e installazione .....	8
1.3.8 Specifiche delle applicazioni utente .....	8
1.3.9 Sviluppo delle applicazioni utente.....	9
1.3.10 Integrazione .....	9
1.3.11 Manutenzione e crescita .....	9
1.3.12 Gestione del progetto.....	9
1.4 Il Modello Multidimensionale .....	10
1.4.1 Dimensioni .....	10
1.4.2 Attributi e gerarchie.....	11
1.4.3 Fatti e misure .....	11
1.5 Implementazione del modello multidimensionale.....	12
1.5.1 I Database Multidimensionali.....	13
1.5.2 I database relazionali .....	16
1.6 Le differenti architetture di un sistema OLAP.....	17
1.6.1 L'architettura MOLAP .....	17
1.6.2 L'architettura ROLAP .....	18
1.6.3 L'architettura HOLAP .....	19
1.6.4 L'architettura DOLAP .....	19
Capitolo 2. La Metodologia Agile .....	21
2.1 Concetti preliminari .....	21
2.1.1 I modelli di progettazione.....	21
2.2 Introduzione all'Agile.....	24

2.2.1	Principi su cui si fondano le tecniche Agile .....	24
2.3	Sviluppare in Agile .....	25
Capitolo 3. Lo strumento di sviluppo: QlikView .....		30
3.1	QlikView in breve.....	30
3.1.1	Struttura di QlikView .....	30
3.1.2	L'architettura di QlikView .....	31
3.2	QlikView Best practices .....	36
3.2.1	Segmentazione delle cartelle di progetto.....	36
3.2.2	Flusso base di un progetto .....	41
Capitolo 4. Il caso di studio: Amplifon DE .....		42
4.1	Descrizione Funzionale.....	43
4.1.1	Scopo del progetto .....	43
4.1.2	Il processo di business .....	44
4.1.3	Il modello analitico .....	49
4.1.4	Architettura del sistema .....	56
4.1.5	End User Interface .....	58
4.2	Analisi dello sviluppo progettuale .....	62
4.2.1	Stima delle attività.....	62
4.2.2	Il modello applicato (incrementale).....	64
4.2.3	I soggetti coinvolti .....	65
4.3	Sviluppo del sistema .....	68
4.3.1	Solution Analysis.....	69
4.3.2	Modeling DataMart .....	72
4.3.3	Staging Area .....	74
4.3.4	Metadata Definition.....	80
4.3.5	Developing UAT Reports.....	84
4.3.6	Technical Documentation.....	93
4.3.7	Formation .....	95
Capitolo 5. Evoluzioni (CR Sales) .....		97
5.1	Evoluzioni Funzionali.....	97
5.2	Analisi e Sviluppo CR Sales .....	100
Capitolo 6. Conclusioni .....		105
6.1.1	Miglioramenti della soluzione sviluppata .....	105
6.1.2	Competenze acquisite .....	107
Bibliografia.....		109



## Introduzione

Questa tesi si propone come obiettivo di descrivere la progettazione e lo sviluppo di un sistema di supporto alle decisioni, partendo dalla costruzione del data warehouse, fino alla costruzione dell'applicazione vera e propria. Per raggiungere questo obiettivo si è intrapresa una strada nuova, basata su un approccio progettuale recente e comunemente detto: *Metodologie Lean-Agile*. Queste tecniche di progettazione permettono di sviluppare sistemi informativi secondo logiche snelle, efficaci ed efficienti. Lo scopo è quello di raggiungere quello che dovrebbe essere il valore massimo per un'azienda: la piena soddisfazione del cliente.

Generalmente l'approccio base di un progetto è quello di affrontare lo sviluppo dall'inizio alla fine senza avere interazioni col cliente o avendone solo in minima parte. Questo comporta spesso un aumento dei tempi e dei costi dei progetti. Tramite l'approccio Agile, invece, si riescono a superare questi ostacoli. Infatti le continue interazioni col cliente portano ad avere una maggiore reattività alle richieste dello stesso, così da poter raggiungere la chiusura del progetto nei termini concordati e con grande soddisfazione dell'azienda medesima.

Per dimostrare come questo sia possibile viene presentato un caso di studio reale.

Come caso di studio è stato svolto un progetto per il reparto *Marketing di Amplifon Germany* (AMPMDE). Nell'elaborato dimostrerò come ho portato avanti la progettazione e lo sviluppo del sistema, che è in grado di fornire in tempi brevi, informazioni strategiche agli utilizzatori. Alla base del suddetto sistema ho progettato e costruito un Data warehouse in grado di rispondere a tutte le funzionalità previste. Il Data warehouse contiene le informazioni che successivamente vengono rese fruibili tramite un'applicazione che fornisce una *dashboard* (insieme di report: documenti in grado di presentare i dati trasformati, formattati ed organizzati in base alle specifiche aziendali, attraverso filtraggi e selezione dei dati da un database). Il sistema è stato sviluppato in tecnologia QlikView, uno degli strumenti di Business Intelligence attualmente più utilizzati per la realizzazione di *dashboard*.

Oggi il sistema è in produzione, e viene utilizzato da tutti gli utenti del reparto Marketing.

*Nella descrizione dell'elaborato sono stati mascherati tutti i dati, per motivi di privacy aziendale, in quanto sono coperti dal diritto di segretezza.*

Di seguito vengono descritti in breve i sei capitoli in cui l'elaborato si suddivide:

Nella prima parte (Capitolo 1), introduttiva, saranno rapidamente introdotti i sistemi di data Warehousing, le procedure per lo sviluppo di un database multidimensionale e le possibili soluzioni adottabili.

Nella seconda parte (Capitolo 2), verranno affrontate le tematiche riguardanti le possibili strategie di sviluppo, e il confronto con quella adottata: Lean-Agile.

La terza parte (Capitolo 3) descriverò in breve il software di riferimento utilizzato, QlikView, e le best practices adottate per lo sviluppo del sistema.

La quarta parte (Capitolo 4) è legata allo sviluppo. Si focalizzerà sullo sviluppo dell'applicazione nelle sue vari fasi progettuali, partendo dalle prime fasi (progettazione e modellazione del data warehouse), passando per la costruzione dell'applicazione, la *Dashboard* per il reparto marketing, e terminando con la fase di *Training*. Le fasi seguono l'esatta evoluzione del progetto, e sono organizzate seguendo l'approccio *Agile*.

Nella quinta parte (Capitolo 5), descriverò le evoluzioni richieste in corso d'opera, di cui dimostrerò la capacità e la flessibilità con sono riuscito a rimediare tramite una corretta gestione della soluzione proposta.

Nell'ultima parte (Capitolo 6) discuterò brevemente dei vantaggi ottenuti da parte dei clienti, dimostrando le differenze fra la soluzione precedentemente adottata e quella successivamente sviluppata, in termini di rapidità e immediatezza di analisi. In conclusione, dimostrerò come, seguendo procedure di Best practices, Iconsulting abbia acquisito la capacità di affrontare progetti in QlikView in modo molto più efficace ed efficiente.



# Capitolo 1. I sistemi di data warehousing

## 1.1 Il Data Warehouse

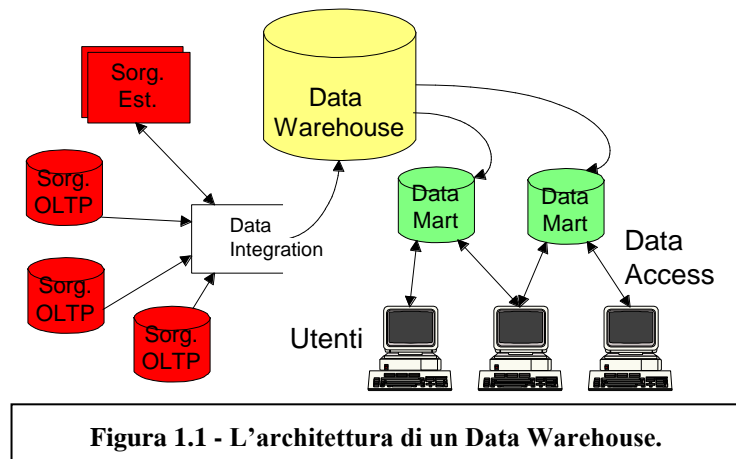
In questi ultimi anni le aziende hanno cambiato radicalmente il loro modo di concepire, progettare ed utilizzare i sistemi informativi. La principale innovazione apportata è stata quella di separare le attività di tipo operativo o attività transazionali (*transactional processing*) da quelle di analisi dei dati e di pianificazione (*analytical processing*). Questo processo di separazione ha portato alla nascita di sistemi dedicati alle attività di analisi dei dati che vengono chiamati *Sistemi di supporto alle decisioni* oppure, utilizzando la terminologia inglese, *Decision Support System* (DSS).

Una definizione più formale di questa classe di sistemi è la seguente: “Un sistema di supporto alle decisioni è un sistema in grado di fornire chiare informazioni agli utenti, in modo che essi possano analizzare dettagliatamente una situazione e prendere facilmente le opportune decisioni sulle azioni da intraprendere.” In altre parole, un DSS è un sistema informativo dedicato ad aiutare i responsabili del business in tutta la gestione d’impresa, in modo da permettere decisioni più veloci, più mirate e più efficaci. Questo sistema deve quindi avere un unico obiettivo: contribuire ad incrementare le performance economiche di un’impresa, attraverso la fornitura di informazioni strategiche in modo semplice e veloce. Un database analitico è pensato per gestire grandi volumi di dati, tipicamente in sola lettura, al fine di produrre le informazioni necessarie ai responsabili di un’organizzazione per prendere le decisioni. Chiameremo allora Data Warehouse “il database analitico che costituisce le fondamenta di un sistema di supporto alle decisioni”.



## L'architettura di un Data Warehouse

La figura 1.1 mostra il flusso, l'organizzazione e l'utilizzo dei dati all'interno di un data warehouse. Il processo di *Data Integration* si occupa di estrarre i dati dalle varie sorgenti OLTP presenti, di trasformarli e organizzarli opportunamente secondo regole ben definite, e di caricarli all'interno del data warehouse. Questo processo viene anche chiamato *Extraction, Transformation & Loading*.



Il data warehouse rappresenta quindi una sorgente di dati secondaria, popolata dai sistemi OLTP esistenti e da altre fonti dati esterne, al fine di facilitare i processi di analisi e di pianificazione.

### 1.1.1 Il Data Mart

Spesso il data warehouse viene partizionato in sottoinsiemi logici chiamati Data Mart. Più formalmente: “Un Data Mart è un database analitico ottenuto come sottoinsieme dei dati presenti nel DW primario, progettato per soddisfare le esigenze di una specifica funzione del business in quanto contenente l'insieme delle informazioni rilevanti per una particolare divisione dell'azienda o una particolare categoria di soggetti.”

Le principali metodologie di progettazione di un data warehouse si basano su uno dei seguenti approcci **Errore. L'origine riferimento non è stata trovata.:**

- *Top-Down*. Questo approccio prevede prima la costruzione del data warehouse, poi l'aggregazione e l'esportazione dei suoi dati verso i più specifici data mart. In

questo modo si dà al progetto una visione globale del business ma si rischia di dover richiedere all'utente molto tempo e molte risorse prima di ottenere dei risultati tangibili.

- *Bottom-Up*. Seguendo invece l'approccio opposto, si inizia la costruzione del data warehouse aziendale in modo scalabile. Questo significa che il progetto inizia focalizzando l'attenzione su specifiche aree del business, per costruire i relativi data mart, che successivamente vengono integrati per formare il data warehouse. E' importante porre particolare attenzione alla scelta dei primi data mart da realizzare, che dovranno essere quelli più interessanti per il particolare business; inoltre è necessario seguire un modello standard per la progettazione dei singoli data mart, stabilito considerando l'intero business nel suo complesso.

### **1.1.2 I metadati**

La definizione anglosassone di metadati è "data about data". Usando una terminologia più esplicativa, possiamo definire i metadati come "dati che descrivono altri dati", cioè, il loro contenuto fornisce informazioni sulle strutture dei dati e sulle rispettive relazioni esistenti all'interno di un singolo database o fra più database.

**L'origine riferimento non è stata trovata..**

In un DSS esistono due tipi di metadati. Il primo tipo è composto dai data integration metadata e contiene i metadati associati alle informazioni provenienti dalle varie sorgenti OLTP. Questi metadati vengono utilizzati durante il processo di estrazione, trasformazione e caricamento (processo ETL). Il secondo tipo è invece composto dai DSS metadata. Questi metadati realizzano il binding delle informazioni contenute nel warehouse verso le applicazioni utente, nascondendo all'utente finale i dettagli implementativi del warehouse e presentandogli invece una interfaccia di lavoro che opera direttamente con termini propri del business.

### **1.1.3 Accesso ai dati: tipologie di applicazioni.**

Vediamo brevemente come è possibile utilizzare le informazioni contenute all'interno di un data warehouse per le necessità aziendali.

- *Query & Report*. L'utente ha la possibilità di accedere ai dati contenuti nel data warehouse direttamente dal proprio pc in maniera semplice ed intuitiva. Esistono diversi strumenti di reportistica più o meno flessibili che offrono la possibilità all'utente di generare dinamicamente i report che gli occorrono per le proprie analisi.
- *Applicazioni Olap*. E' il metodo più diffuso ed efficace di accesso ai dati. La direzione ha la possibilità di effettuare analisi avanzate su grandi moli di dati, di crearsi un proprio percorso dinamico di visualizzazione dei report. Grazie a queste applicazioni si possono risolvere efficacemente molte funzioni del business: oltre all'analisi storica dei dati nasce anche la possibilità di effettuare delle previsioni sugli andamenti futuri (*what-if analysis*). Le principali tipologie di strumenti che permettono questo tipo di applicazioni sono *Multidimensional Database* che memorizzano le informazioni direttamente in formato multidimensionale e gli *Advanced decision support tools* che permettono l'analisi multidimensionale dei dati direttamente su un database relazionale.
- *Data Mining*. Con questo termine si indica un processo che, utilizzando delle tecniche statistiche, permette la scoperta di deboli relazioni esistenti all'interno di un grande insieme di dati. Successivamente, basandosi sui vari legami messi in luce, è possibile costruire dei modelli per elaborare previsioni sul comportamento del sistema osservato. Le applicazioni di data mining permettono, quindi, di estrapolare informazioni da un insieme di dati e di trasformarle in conoscenza.

## 1.2 Data Warehouse vs. Database OLTP.

I classici database transazionali differiscono dai database analitici sia nella progettazione che nelle funzionalità. Le differenze più macroscopiche riguardano:

- **L'organizzazione dei dati**. I dati all'interno di un sistema OLTP sono orientati al tipo di applicazione e spesso risultano eterogenei, ovvero i diversi sistemi OLTP presenti all'interno di un'organizzazione contengono differenti tipi di dati. All'interno di un data warehouse, invece, tutti i tipi di dati vengono integrati in un

unico database e vengono organizzati secondo le dimensioni (o prospettive) di analisi del business: sono quindi rivolti al soggetto di interesse.

- **L'integrazione dei dati.** I dati contenuti in differenti sistemi OLTP non sono integrati fra loro. Il data warehouse rappresenta, invece, un contenitore di dati perfettamente integrati. All'interno, infatti, sono state standardizzate le strutture delle chiavi, i nomi e il formato dei file.
- **Dati storici.** Spesso, i sistemi OLTP mantengono in linea i soli dati correnti. In un data warehouse, invece, vengono contenuti tipicamente dai 2 ai 5 anni di dati storici, al fine di fornire supporto a particolari tipi di analisi temporali, come la visualizzazione dei trend di determinate misure lungo il tempo, o la comparazione fra periodi di tempo (ad esempio, anno corrente vs. anno passato).
- **L'accesso ai dati.** Nei sistemi OLTP l'accesso ai dati avviene attraverso operazioni di interrogazione, aggiornamento, inserimento e cancellazione che coinvolgono una piccola quantità di dati. Il controllo della consistenza in questi sistemi è effettuato a livello di singola transazione. In un data warehouse il tipo di accesso ai dati è rappresentato da operazioni piuttosto pesanti, tipicamente in sola lettura e coinvolgenti una grande, a volte enorme, quantità di dati. I dati contenuti nel warehouse non sono costantemente aggiornati, ma vengono caricati ad intervalli periodici di tempo. Il controllo della consistenza dei dati viene effettuato solo durante questa operazione di caricamento.
- **L'utilizzo delle risorse di sistema.** Il livello di utilizzazione di un sistema transazionale presenta dei massimi e dei minimi, ma tendenzialmente il carico del sistema si mantiene costante. Il data warehouse, invece, o utilizza l'intero sistema al 100% oppure non lo utilizza affatto. Questo profilo di utilizzazione nasce dal fatto che tipicamente l'utilizzo dei DSS consiste nell'esecuzione di grandi interrogazioni, molto impegnative per il sistema, e di lunghe pause per analizzare i risultati ottenuti.

## 1.3 Il ciclo di vita di un Data Warehouse.

Le numerose caratteristiche che differenziano i Data Warehouse dai sistemi transazionali fanno sì che anche la fase di progettazione del database richieda sostanziali variazioni.

In questa sezione, vedremo una panoramica di tutte le attività che sovrintendono la creazione e il mantenimento di un Data Warehouse. L'insieme di queste attività prende il nome di ciclo di vita ed è rappresentato in figura 1.2 .

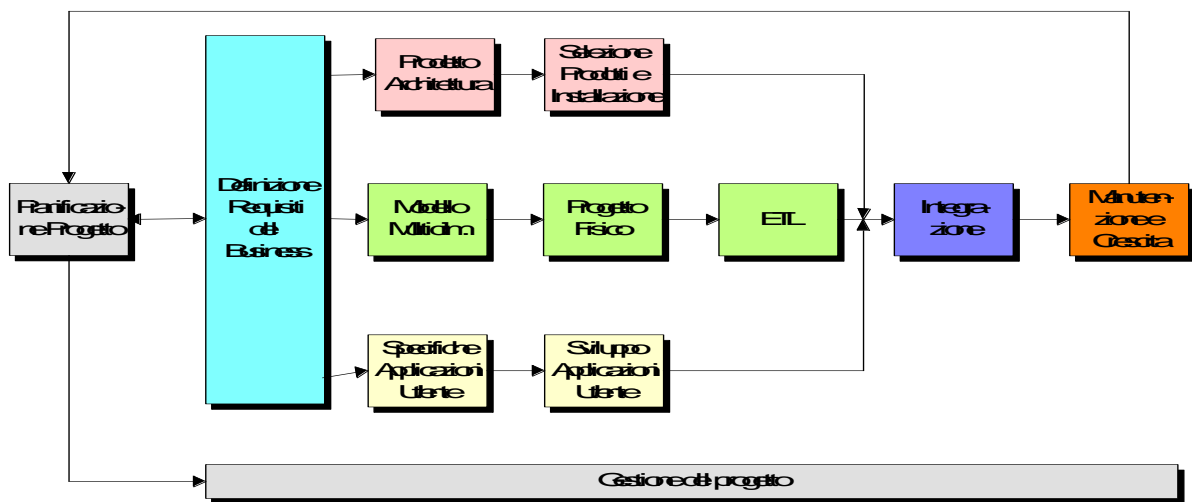


Figura 1.2 – Il ciclo di vita di un sistema di supporto alle decisioni.

### 1.3.1 Pianificazione del Progetto

Il ciclo di vita comincia, naturalmente, con la pianificazione dell'intero progetto. Pianificare significa definire le attività del progetto, stabilire le risorse da impiegare e lo staff di risorse umane da utilizzare. Il risultato di questo lavoro sarà un documento strutturato che identifica ed organizza tutte le attività coinvolte nel ciclo di vita del sistema e le relative parti chiamate in causa. Notiamo, inoltre, che la pianificazione è strettamente dipendente dai requisiti del business da modellare.

### **1.3.2 Definizione dei requisiti del Business**

La probabilità di successo di un DSS è tanto più elevata quanto migliore è stata la comprensione dei requisiti del business e degli utenti finali. Chi progetta un data warehouse, infatti, deve capire perfettamente i fattori chiave che guidano il business, in modo da inquadrare inequivocabilmente quali siano questi requisiti.

### **1.3.3 Il modello multidimensionale**

La definizione dei requisiti permette di definire il modello dei dati necessario per soddisfare le esigenze richieste dagli utenti. E' importante notare che la progettazione del modello dei dati differisce notevolmente rispetto all'approccio adottato per i sistemi transazionali. Questo processo si compone di una serie di passi che verranno approfonditi nei capitoli successivi dedicati allo sviluppo del progetto.

### **1.3.4 Il progetto fisico**

Il progetto fisico del database focalizza l'attenzione sulla definizione delle strutture fisiche necessarie per supportare il progetto logico definito nel punto precedente. In pratica, è necessario definire alcuni aspetti:

- una *convenzione standard per i nomi* che renda il progetto chiaro e leggibile
- la *configurazione dell'ambiente del database* e le strategie di *indicizzazione e di partizionamento* per il raggiungimento delle prestazioni massime del sistema.

### **1.3.5 Extraction, Transformation & Loading (ETL)**

Questa attività, spesso sottostimata, si compone di tre processi fondamentali: l'estrazione dei dati dalle varie fonti OLTP, la loro trasformazione e il caricamento nel data warehouse.

Il processo di trasformazione è particolarmente delicato perché deve garantire la qualità dei dati del warehouse. Il controllo della qualità dovrà essere effettuato prima di rendere i dati disponibili agli utenti finali attraverso una serie di report statistici sui dati appena caricati. Generalmente vengono realizzati due processi ETL: il primo utilizzato per popolare la prima volta il warehouse, il secondo per

caricare incrementalmente e ad intervalli regolari le nuove informazioni non ancora presenti nel warehouse.

### **1.3.6 Il progetto dell'architettura tecnologica**

Un DSS richiede l'integrazione di numerose tecnologie. Per progettare l'architettura tecnologica bisogna considerare alcuni fattori fondamentali, quali i *requisiti del business*, che indicano le linee guida del sistema, ma anche l'*architettura tecnologica in essere*, fondamentale per capire l'integrazione delle nuove tecnologie con quelle esistenti, e le *strategie pianificate nell'area tecnologica* che definiscono i budget disponibili per l'acquisto di nuove tecnologie.

### **1.3.7 Selezione dei prodotti e installazione**

Utilizzando i risultati del punto precedente, bisogna procedere alla selezione dei vari componenti dell'architettura stabilita. Questo si traduce nella valutazione dei seguenti punti chiave:

- Le *piattaforme hardware*, necessarie per l'implementazione del sistema DSS.
- Il *database management system*, che dovrà contenere e gestire la base dati.
- Gli *strumenti ETL*, necessari per caricare le informazioni nel DSS.
- Gli *strumenti di accesso*, utilizzati per visualizzare e manipolare i dati contenuti nel warehouse.
- 

### **1.3.8 Specifiche delle applicazioni utente**

E' raccomandabile definire un set di applicazioni utente standard, poiché non tutti gli utenti hanno l'esigenza di avere un accesso personalizzato al DSS. Le specifiche delle applicazioni sono composte sostanzialmente da documenti che descrivono il modello dei report richiesti, i parametri utente e i calcoli richiesti.

### **1.3.9 Sviluppo delle applicazioni utente**

Seguendo le specifiche raccolte si può sviluppare un'applicazione in grado di fornire semplici strumenti all'utente finale per modificare, secondo le proprie esigenze, i report esistenti.

### **1.3.10 Integrazione**

Con questo termine si rappresenta la convergenza e la perfetta integrazione delle tecnologie, dei dati e delle applicazioni utente, in modo che siano pienamente utilizzabili dai desktop dell'utente.

### **1.3.11 Manutenzione e crescita**

L'attività di manutenzione coinvolge una serie di processi:

- *Il supporto e la formazione degli utenti.* Per permettere l'appropriato utilizzo delle nuove risorse disponibili.
- *La manutenzione delle applicazioni.* I requisiti di un business sono molto dinamici e le richieste effettuate dagli utenti cambiano nel tempo.
- *La supervisione del DSS.* E' necessario mantenere sotto controllo il comportamento e le prestazioni del sistema.
- *La realizzazione di una rete di comunicazione.* Si vuole ottenere la massima integrazione fra tutti i partecipanti del progetto.

Seguendo questo tipo di approccio si ottiene un sistema modulare ed aperto in grado di crescere ed evolversi nel tempo.

### **1.3.12 Gestione del progetto**

Gestire un progetto di un sistema DSS è una tipica attività di *project management*. Questo significa, sostanzialmente, controllare che le attività coinvolte nel ciclo di vita procedano rispettando i tempi previsti e rimangano sincronizzate fra loro.



## 1.4 Il Modello Multidimensionale

La modellazione multidimensionale è la tecnica più utilizzata per dare una rappresentazione grafica ai dati contenuti in un sistema di supporto alle decisioni. Infatti, attraverso questo modello, è possibile identificare immediatamente il raggio d'azione del data warehouse o del data mart. Inoltre, il modello multidimensionale dei dati costituisce la base per la progettazione dello schema logico del database evidenziando le dimensioni e i fatti del business.

### 1.4.1 Dimensioni

Possiamo definire il termine *dimensione* come “una prospettiva di analisi dei dati di un business”. Attraverso le dimensioni, sarà quindi possibile visualizzare e studiare gli stessi tipi di dati secondo diversi punti di vista. I dati vengono organizzati secondo differenti prospettive, e quindi possono essere rappresentati mediante un *cubo* avente tante dimensioni quante sono le prospettive di analisi stabilite. Le dimensioni rappresenteranno, quindi, gli spigoli di questo cubo.

In figura 1.3, viene rappresentato un cubo a tre dimensioni. In questo caso, ogni singola cella rappresenta la quantità venduta di un dato prodotto, in un dato negozio e in un determinato periodo di tempo.

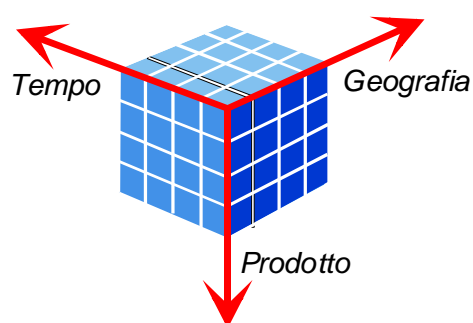


Figura 1.3 – Un esempio di cubo a 3 dimensioni.

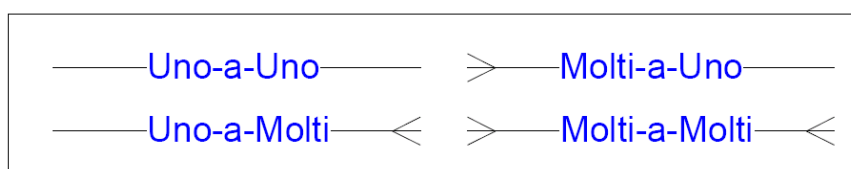
I valori contenuti all'interno di una dimensione sono chiamati *elementi della dimensione* e fungono da coordinate per identificare le singole celle del cubo dei dati.

## 1.4.2 Attributi e gerarchie

Un attributo rappresenta un raggruppamento logico di alcuni elementi appartenenti alla stessa dimensione. Gli attributi, quindi, sono classi di elementi che permettono all'utente di selezionare i dati attraverso specifiche caratteristiche. Esempi di attributi sono: la marca, il colore, la taglia, etc.

Gli elementi di attributi relativi ad una stessa dimensione sono legati da diversi tipi di relazioni:

- *uno-a-uno*. Ad un elemento padre corrisponde un solo elemento figlio.
- *uno-a-molti*. Un elemento padre può avere uno o più elementi figli.
- *molti-a-uno*. Un elemento figlio può avere uno o più padri.
- *molti-a-molti*. Un elemento figlio può avere uno o più padri e un elemento padre può avere uno o più figli.



**Figura 1.4 - Rappresentazione relazioni**

Queste relazioni esistenti fra gli attributi costituiscono una gerarchia. Per definirla basterà, semplicemente, stabilire gli attributi che la compongono e le relazioni uno-a-molti esistenti fra loro. Le gerarchie sono fondamentali per la navigazione dei dati. Infatti, permettono di specificare secondo quale modalità i dati devono essere aggregati salendo verso un livello superiore (operazione di *Roll-Up*) e in che modo avviene l'esplosione dei dati scendendo verso un livello inferiore (operazione di *Drill-Down*).

## 1.4.3 Fatti e misure

I fatti non sono altro che gli eventi reali di un business. Ogni singolo fatto può essere organizzato lungo una o più dimensioni. L'insieme delle dimensioni su cui viene organizzato si chiama *dimensionalità* di quel fatto. Nel paradigma multidimensionale, ogni singolo fatto può essere pensato come un *cubo di dati* aventi tanti spigoli quante

sono le sue dimensioni. Un fatto deve sempre avere degli aspetti dinamici che lo rendano interessante per il processo decisionale. Inoltre può essere caratterizzato da una o più misure. Una *misura* è una proprietà numerica di un fatto e ne descrive un aspetto quantitativo di interesse per l'analisi. **Errore. L'origine riferimento non è stata trovata.** Ad esempio, per il fatto *Vendite* le misure tipiche possono essere la quantità venduta o il valore del venduto.

Nel modello multidimensionale, i fatti vengono rappresentati mediante dei cubi e la loro dimensionalità viene rappresentata da frecce che li collegano con le rispettive dimensioni. In figura 1.5 viene rappresentato un semplice modello multidimensionale. In questo esempio la granularità dei fatti è definita dalla tripla (Item, Negozio, Giorno).

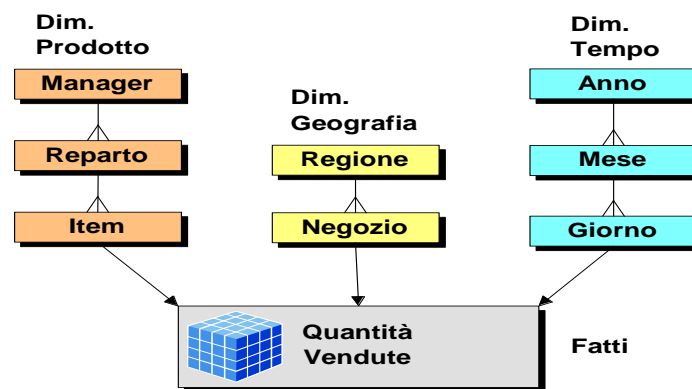


Figura 1.5 – Un semplice modello multidimensionale.

## 1.5 Implementazione del modello multidimensionale

Una volta definito il modello multidimensionale del business, bisognerà implementarlo fisicamente su un sistema. Questo comporta la scelta della tecnologia di supporto, il disegno logico del database e la sua realizzazione fisica. Nei prossimi paragrafi verranno introdotte le due differenti tecnologie per la memorizzazione dei dati e il relativo progetto logico del database.

### 1.5.1 I Database Multidimensionali

L'ultimo passo nell'evoluzione dei database è il database multidimensionale. La filosofia su cui si basa questo sistema è quella di realizzare una corrispondenza biunivoca fra come le persone vedono i dati del loro business e come questi dati vengono memorizzati in un sistema informativo. Utilizzando questa tecnologia, la progettazione logica del database è praticamente inesistente, in quanto esiste una relazione uno-a-uno fra il modello multidimensionale e il disegno del database. Infatti, questo sistema software permette di definire gli stessi oggetti che abbiamo visto nel modello multidimensionale: dimensioni, attributi, relazioni, livelli, gerarchie e fatti.

La struttura fondamentale di memorizzazione di un database multidimensionale è l'*array* scelto per sostituire le tabelle relazionali, da cui per altro si differenziano per alcuni aspetti fondamentali che ora andremo ad analizzare.

I dati in un database relazionale vengono memorizzati sotto forma di tabelle, collegate fra loro attraverso chiavi. All'interno di ogni tabella le informazioni sono organizzate in records. Ogni record corrisponde ad una riga della tabella ed è suddiviso in campi che costituiscono le colonne della tabella stessa. In figura 1.6 è rappresentato un esempio di tabella relazionale.

<b>MODEL</b>	<b>COLOR</b>	<b>SALES VOLUME</b>
Mini Van	Blue	6
Mini Van	Red	5
Mini Van	White	4
Sport Coupe	Blue	3
Sport Coupe	Red	5
Sport Coupe	White	5
Sedan	Blue	4
Sedan	Red	3
Sedan	White	2

**Figura 1.6 – Una tipica tabella relazionale.**

Come possiamo notare, la tabella contiene nove records, ciascuno dei quali è composto da tre campi : model, color e sales volume. Per rappresentare le stesse informazioni, un database multidimensionale utilizzerà l'array bidimensionale mostrato in figura 1.7.

## Sales Volumes

MODEL	Mini Van	6	5	4
	Coupe	3	5	5
	Sedan	4	3	2
		Blue	Red	White
		COLOR		

**Figura 1.7 – Il corrispondente array bidimensionale.**

Guardando la figura 1.7, si può vedere immediatamente che ci sono esattamente due dimensioni formate da tre elementi ciascuna, cosa molto più difficile da notare su una tabella, soprattutto al crescere delle dimensioni e degli elementi. Inoltre, l'array multidimensionale presenta un più alto livello di organizzazione dei dati che permette una visualizzazione ed una manipolazione molto più logica per l'utente finale.

Questo tipo di organizzazione permette anche grossi vantaggi in termini di performance. Consideriamo ad esempio un cubo di 3 dimensioni contenenti ciascuna 10 elementi. L'equivalente tabella relazionale deve contenere  $10 \times 10 \times 10 = 1000$  records. Se un utente vuole conoscere il valore di una determinata cella, il sistema relazionale deve scandire, nel caso peggiore, tutti i 1000 records, mentre il sistema multidimensionale deve cercare solo lungo le tre dimensioni per identificare le coordinate del dato. Questo si traduce nella scansione di un massimo di 30 elementi.

Una nota dolente per i sistemi multidimensionali è rappresentata, invece, dalla *sparsità*. Infatti, se i vari record di una tabella relazionale non godono di interrelazioni, cioè i campi corrispondenti alle dimensioni non sono composti da pochi elementi ripetuti, quello che si ottiene, mappandola su un database multidimensionale, è un *array sparso* (molte celle sono vuote) poco prestante in termini di performance e utilizzo di spazio.

Questa tipologia di sistemi è stata concepita con l'intento di massimizzare le performance dei processi analitici realizzando una corrispondenza fra come l'utente pensa ed utilizza i dati di un business e come il sistema li gestisce effettivamente.

Esistono numerose caratteristiche proprie di un database multidimensionale (MDDDB), vediamole brevemente.

- *Analisi multidimensionali* – In particolare è banale immaginare come le operazioni di rotazione e di selezione avvengano naturalmente su questi database senza la necessità di riorganizzare i dati e con ottime prestazioni.
- *Gestione delle gerarchie* - Le gerarchie sono viste come differenti livelli di aggregazione collegati attraverso relazioni uno-a-molti. I dati aggregati ai vari livelli vengono memorizzati nel database multidimensionale in modo da non dover essere calcolati a run-time. Per ottenere lo stesso effetto su un sistema relazionale, occorre rilanciare la query ogni volta che cambia il livello della gerarchia.
- *Linguaggio di interrogazione* – L'alto grado di strutturazione che si ottiene con un MDDDB si traduce anche in un semplice ed efficiente linguaggio di interrogazione. Non solo il linguaggio è più intuitivo, ma l'output della query è molto più significativo per l'utente essendo in formato multidimensionale.
- *Accesso Multiutente* – Uno dei vantaggi chiave di un MDDDB consiste nella possibilità di gestire meglio l'accesso concorrente alle stesse informazioni da parte di più utenti. Ogni utente è in grado di definire agevolmente la propria vista dei dati e di compiere le relative operazioni senza interferire con gli altri.
- *Calcoli Multidimensionali* – Un MDDDB permette di eseguire calcoli aritmetici sui dati in modo semplice ed efficiente.
- *La Dimensione Tempo* – In molti MDDDB il tempo è una dimensione predefinita, con gerarchie predefinite. Quindi esistono funzioni preconfezionate che operano su serie temporali in modo molto efficiente.
- *Sparsità* – Gli sparse array sono cubi con poche celle popolate. Le celle vuote, cioè quelle che non contengono dati, sono contrassegnate da uno speciale valore NA (*Not Available*). Un buon MDDDB deve trattare in modo intelligente i valori NA utilizzando tecniche di compressione del DB che permettano di ottenere ottime performance riducendo lo spazio di memorizzazione richiesto.

## 1.5.2 I database relazionali

Nelle sezioni precedenti abbiamo visto che, utilizzando un database multidimensionale, non esiste la necessità di definire un progetto logico della base dati perché sussiste una corrispondenza biunivoca col modello multidimensionale.

La situazione si presenta decisamente differente se scegliamo come supporto di memorizzazione un sistema relazionale. In questo caso, infatti, occorre utilizzare una tecnica di progettazione logica espressamente pensata per rappresentare nel miglior modo possibile strutture dati multidimensionali. Il risultato di questo processo sarà uno schema relazionale composto da differenti tipi di tabelle che rappresentano gli oggetti caratteristici del modello multidimensionale: una tabella dei fatti contenente le misure richieste e una tabella per ogni dimensione che conterrà tutti gli elementi della gerarchia.

Il modello finale che si ottiene è detto Star Schema proprio per la sua caratteristica forma a stella con la tabella dei fatti al centro e le tabelle delle dimensioni attorno. La figura 1.8 ne riporta un esempio.

La caratteristica dello *star schema* è di essere pesantemente denormalizzato e di presentare, quindi, una grande ridondanza dei dati. L'operazione di denormalizzazione permette però di ridurre il numero di join richiesti per soddisfare una interrogazione.

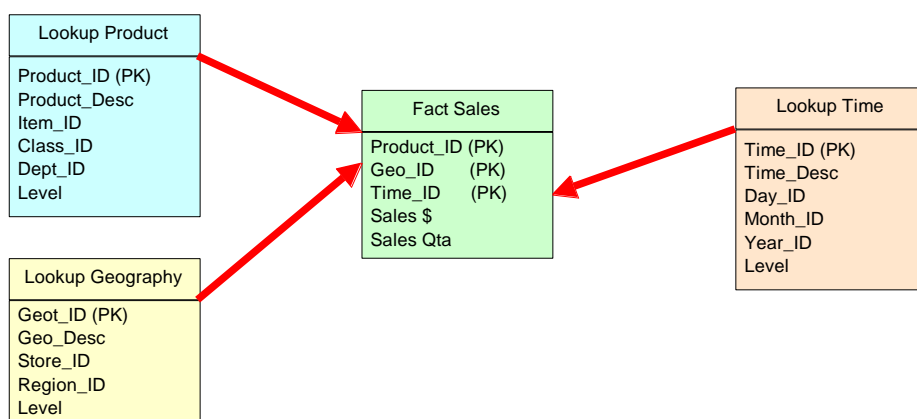


Figura 1.8 – Lo star schema.

Questo schema viene realizzato con alcune varianti in base alle necessità del business. Innanzi tutto è possibile mantenere nella fact table solo i dati atomici o anche quelli aggregati riducendo i tempi di roll-up ma aumentando il numero di record

memorizzati. Inoltre, lavorando sulle dimension table si ottengono degli schemi via via più normalizzati detti snowflake schema che riducono le dimensioni delle varie tabelle aumentando però il numero di join necessari. In pratica si implementa per ogni livello della gerarchia una diversa tabella; i vari livelli sono collegati con l'uso di foreign key e solo il livello più basso è legato alla fact table che dovrà quindi mantenere sempre dati non aggregati.

## 1.6 Le differenti architetture di un sistema OLAP

Nella sezione precedente abbiamo visto come sia possibile implementare il modello multidimensionale utilizzando un database multidimensionale o un database relazionale. Le diverse tipologie dei prodotti OLAP reperibili sul mercato si differenziano proprio in base al tipo di sistema scelto. E' opportuno notare che ogni tipo di architettura presenta dei punti di forza e delle debolezze rispetto alle altre. Non esiste una scelta ottimale perseguibile in assoluto. La selezione del tipo di architettura influenzerà le performance, la capacità, la funzionalità e la scalabilità del nostro DSS. Vediamo allora, molto sommariamente, le caratteristiche, i relativi vantaggi e gli svantaggi delle quattro principali architetture.

### 1.6.1 L'architettura MOLAP

La generica architettura MOLAP è composta da due livelli come si può notare in figura 1.9.

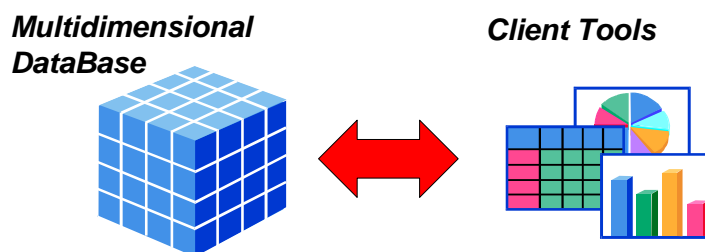


Figura 1.9 – La generica architettura MOLAP.



In questo caso, il database multidimensionale funge sia da struttura di memorizzazione dei dati che da motore OLAP. Infatti, possiede un linguaggio espressamente progettato per effettuare manipolazioni e computazioni multidimensionali.

### **Vantaggi:**

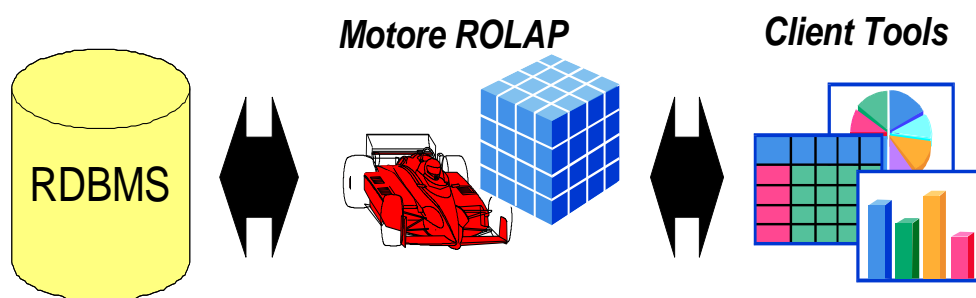
- Ottime performance per l'analisi multidimensionale.
- Permette l'utilizzo di *thin-client* e l'accesso attraverso Internet.
- Supporta analisi predefinite lungo la dimensione tempo.
- I dati analitici sono memorizzati in una struttura persistente, che riduce il carico del sistema per effettuare calcoli e aggregazioni a run-time.

### **Svantaggi:**

- La natura aggregata dei dati memorizzati limita le capacità di Drill-down.
- Mancanza di standard fra i vari prodotti commerciali.
- Costi di gestione e di sviluppo aggiuntivi, essendo una tecnologia specifica per l'analisi.
- Non utilizzabile con grandi volumi di dati atomici (> 10 Gbyte).

## **1.6.2 L'architettura ROLAP**

I prodotti appartenenti a questo settore utilizzano il database relazionale per memorizzare i dati e i relativi metadati. Il motore di calcolo deve essere in grado di tradurre le tipiche operazioni multidimensionali in query SQL e di elaborare successivamente il risultato, per presentarlo in formato multidimensionale. In figura 1.10 viene rappresentata l'architettura tipica di un prodotto ROLAP.



### **Vantaggi:**

- Supporta un grande volume di informazioni base.
- Permette l'utilizzo di *thin-client* e l'accesso attraverso Internet.

**Figura 1.10 – La generica architettura ROLAP.**

### **Svantaggi:**

- Richiede tecniche ad hoc di progettazione del database.
- Le performance sono altamente dipendenti dal tipo di schema utilizzato.
- Non permette di effettuare complesse computazioni finanziarie.

## **1.6.3 L'architettura HOLAP**

L'architettura HOLAP permette l'integrazione dei database multidimensionali con quelli relazionali. Tipicamente, nel database multidimensionale risiedono i dati ad alto livello di aggregazione, mentre in quello relazionale vengono mantenute le informazioni relative ai livelli più dettagliati. In questo modo, quando viene raggiunta la fine di una gerarchia multidimensionale, durante l'operazione di drill-down, l'applicazione ottiene il livello successivo di dettaglio dal DB Relazionale. Questa architettura cerca di combinare i vantaggi delle soluzioni precedenti, diventando però costosa da realizzare. Inoltre richiede tecniche ad hoc di progettazione del database.

## **1.6.4 L'architettura DOLAP**

All'interno di questa categoria rientrano i prodotti OLAP che utilizzano il client sia per memorizzare i dati da analizzare, che per installare un motore di calcolo. Tipicamente questi prodotti hanno funzionalità molto meno sofisticate rispetto ai sistemi OLAP precedenti. Inoltre la mole dei dati che è possibile analizzare è generalmente molto bassa, in quanto limitata dalle caratteristiche del client utilizzato. Nonostante sia semplice da installare e da gestire, economica da implementare e da mantenere

richiede un client sufficientemente potente da potersi creare una propria vista dei dati. La necessità di importare dei dati sul client limita le possibilità di analisi, appesantisce il traffico di rete e rende quindi impossibile l'accesso tramite internet.

# Capitolo 2. La Metodologia Agile

## 2.1 Concetti preliminari

Nell'ingegneria del software, per metodologia agile (o leggera) o metodo agile si intende un particolare metodo per lo sviluppo del software che coinvolge quanto più possibile il committente, ottenendo in tal modo un'elevata reattività alle sue richieste. Esiste un certo numero di tali metodologie, tra le quali vanno citate quelle della Agile alliance, una organizzazione no-profit creata allo scopo di diffonderle.

### 2.1.1 I modelli di progettazione

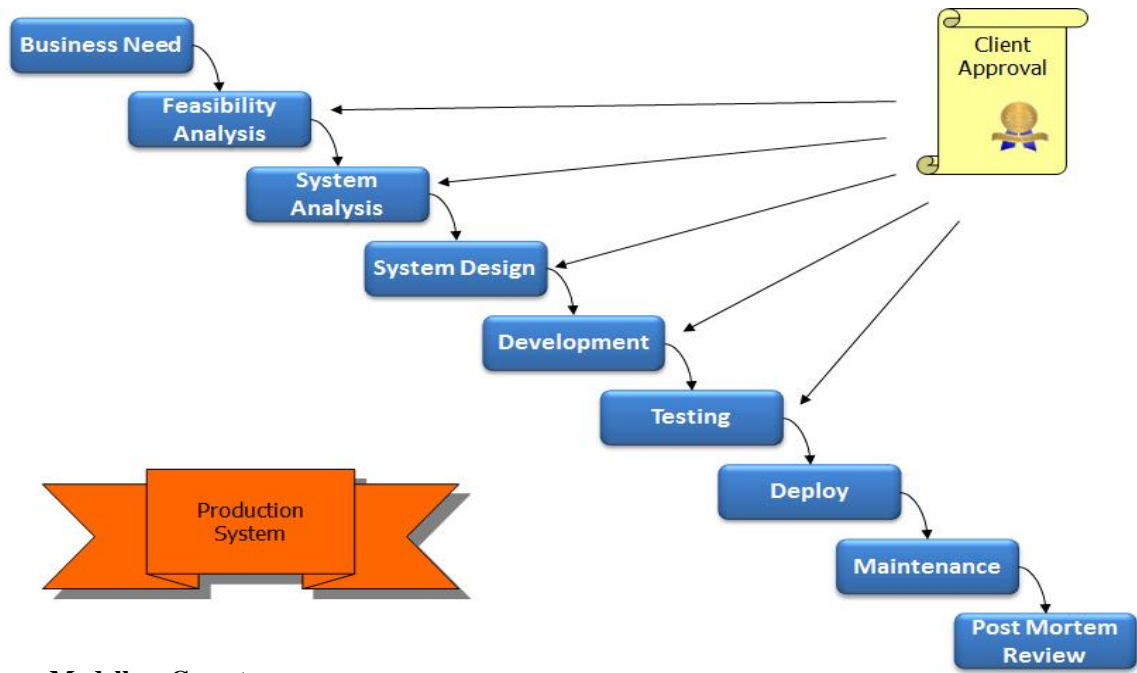
- **Modello a Cascata**

*Pregi del modello a Cascata:*

- ✓ Il processo di sviluppo del software deve essere soggetto a disciplina e pianificazione.
- ✓ L'implementazione del prodotto deve essere rimandata fino a quando non sono perfettamente chiari gli obiettivi.
- ✓ Il maggior pregio di questo metodo di lavoro è certamente la semplificazione del controllo dell'andamento del progetto tramite la suddivisione del ciclo di vita in fasi successive ben definite.

*Limiti del modello a Cascata*

- ✓ Linearità: spesso si hanno cicli di feedback per la correzione degli errori che comportano una revisione molto costosa delle fasi già chiuse, che per la struttura stessa del modello ricade sul cliente; neanche il più perfetto degli utenti è in grado di prevedere tutto subito.
- ✓ Rigidità: il modello a cascata scoraggia i cambiamenti, ma i cambiamenti sono l'assoluta normalità e devono essere gestiti senza eccessivi aumenti di costi a carico del cliente.
- ✓ Monoliticità: tutto il modello è orientato alla singola data di rilascio che spesso si pone a mesi o anni dopo l'inizio della prima fase.

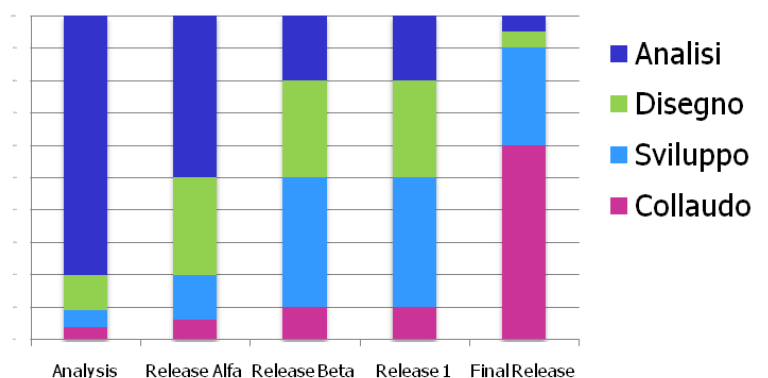


**Modello a Cascata**

- **Modelli incrementali**

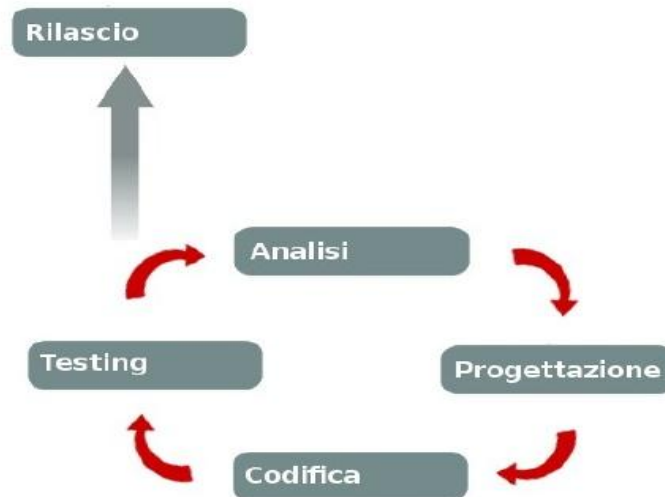
Ogni **iterazione** è un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo **incremento** nelle funzionalità del software: pianificazione (*planning*), analisi dei requisiti, analisi, implementazione, test e documentazione.

- ✓ Maggiore cura nell'analisi dei requisiti
- ✓ Pratiche ad hoc per l'analisi ed il design
- ✓ Fornire una parte del prodotto (o un prototipo) il prima possibile
- ✓ Ogni rilascio segue un piccolo modello a cascata
- ✓ Validare o modificare i requisiti sulla base del rilascio precedente
- ✓ Convergere verso l'applicazione d'insieme richiesta



**Modello incrementale**

## Iterazioni e incremento di valore



Come mostrato in figura, le interazioni prevedono una serie di flussi ciclici continui. Ad ogni ciclo corrisponde il rilascio di una o un'insieme di funzionalità. Il ciclo si compone di 4 fasi principali:

- **Analisi:** fase di analisi dei requisiti e delle funzionalità del sistema o della componente da sviluppare;
- **Progettazione:** viene progettata la funzionalità precedentemente definita in fase di analisi, valutandone la complessità e gli impatti sull'intero sistema.
- **Codifica:** viene generato il codice che implementa le funzionalità, concordate in fase di analisi e progettate in fase progettuale;
- **Testing:** si effettuano i test per valutare l'efficacia del sistema e per assicurarsi che l'informazione fornita sia consistente a quella desiderata.
- **Rilascio:** viene rilasciata la funzionalità all'utente, coinvolgendolo nel processo di controllo ed utilizzo della stessa per cercare di individuare eventuali bug o inconsistenze, e risolvere prima possibile.

## 2.2 Introduzione all'Agile

Il termine Metodologie Agili fu coniato nel 2001 quando il Manifesto Agile è stato formulato.

La gran parte dei metodi agili tenta di ridurre il rischio di fallimento sviluppando il software in finestre di tempo limitate chiamate iterazioni che, in genere, durano qualche settimana. Ogni iterazione è un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software: pianificazione (planning), analisi dei requisiti, progetto, implementazione, test e documentazione.

Anche se il risultato di ogni singola iterazione non ha sufficienti funzionalità da essere considerato completo deve essere rilasciato e, nel susseguirsi delle iterazioni, deve avvicinarsi sempre di più alle richieste del cliente. Alla fine di ogni iterazione il team deve rivalutare le priorità di progetto.

I metodi agili preferiscono la comunicazione in tempo reale, preferibilmente faccia a faccia, a quella scritta (documentazione). Il team agile è composto da tutte le persone necessarie per terminare il progetto software. Come minimo il team deve includere i programmatori ed i loro clienti (con clienti si intendono le persone che definiscono come il prodotto dovrà essere fatto: possono essere dei product manager, dei business analysts, o veramente dei clienti).

### 2.2.1 Principi su cui si fondano le tecniche Agile

L'intento di una programmazione di tipo Agile è la piena soddisfazione del cliente e non solo l'adempimento di un contratto. L'uso di queste metodologie, inoltre, serve ad abbattere i costi di sviluppo del software.

*I principi* su cui si basa una metodologia leggera che segua i punti indicati dall'Agile Manifesto, sono:

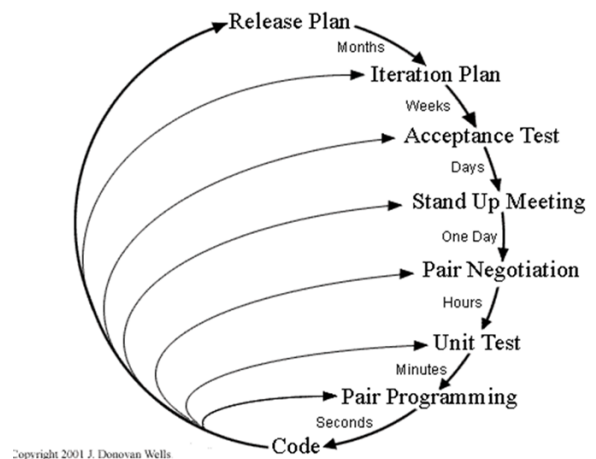
1. La nostra massima priorità è soddisfare il cliente rilasciando software di **valore**, fin da subito e in maniera continua.

2. Accogliamo i **cambiamenti** nei requisiti, anche a stadi avanzati dello sviluppo. I processi agili sfruttano il cambiamento a favore del vantaggio competitivo del cliente.
3. **Consegniamo frequentemente** software funzionante, con cadenza variabile da un paio di settimane a un paio di mesi, preferendo i periodi brevi.
4. Committenti e sviluppatori devono **lavorare insieme** quotidianamente per tutta la durata del progetto.
5. Fondiamo i progetti su **individui motivati**. Diamo loro l'ambiente e il supporto di cui hanno bisogno e confidiamo nella loro capacità di portare il lavoro a termine.
6. Una conversazione **faccia a faccia** è il modo più efficiente e più efficace per comunicare con il team ed all'interno del team.
7. Il software funzionante è il principale metro di **misura di progresso**.
8. I processi agili promuovono uno sviluppo sostenibile. Gli sponsor, gli sviluppatori e gli utenti dovrebbero essere in grado di mantenere indefinitamente un **ritmo costante**.
9. La continua **attenzione all'eccellenza** tecnica e alla buona progettazione esaltano l'agilità.
10. La **semplicità** - l'arte di massimizzare la quantità di lavoro non svolto - è essenziale.
11. Le architetture, i requisiti e la progettazione migliori emergono da team che si **auto-organizzano**.
12. A intervalli regolari **il team riflette** su come diventare più efficace, dopodiché regola e adatta il proprio comportamento di conseguenza.

## 2.3 Sviluppare in Agile



Le singole pratiche applicabili all'interno di una metodologia leggera sono decine e dipendono essenzialmente dalle necessità dell'azienda e dall'approccio del project manager. Nella scelta però bisogna tenere conto delle caratteristiche di ogni pratica per i benefici che apporta e le conseguenze che comporta. Ad esempio, in Extreme Programming, si supplisce alla mancanza assoluta di qualsiasi forma di progettazione e documentazione con lo strettissimo coinvolgimento del cliente nello sviluppo e con la progettazione in coppia.



Le pratiche più diffuse tra cui scegliere sono simili fra di loro e possono essere raggruppate in categorie:

- *Automazione* - Se l'obiettivo delle metodologie leggere è concentrarsi sulla programmazione senza dedicarsi alle attività collaterali, allora possono essere eliminate o automatizzate; la seconda soluzione è migliore perché si può, ad esempio, eliminare la documentazione aumentando il testing, ma non si possono eliminare entrambe; quindi si sceglie che strada si vuole percorrere e si fa in modo da utilizzare strumenti per automatizzare il maggior numero di attività;
- *Comunicazione stretta* - Secondo Alistair Cockburn, probabilmente il primo teorico delle metodologie leggere, questo è l'unico vero aspetto nodale che renda leggera una metodologia. Per comunicazione diretta si intende la comunicazione interpersonale, fra tutti gli attori del progetto, cliente compreso. Ciò serve ad avere una buona analisi dei requisiti ed una proficua collaborazione fra programmatori anche in un ambito di quasi totale assenza di documentazione;
- *Coinvolgimento del cliente* - Il coinvolgimento del cliente è qui indicato singolarmente perché ci sono differenti gradi di coinvolgimento possibili; ad esempio in Extreme Programming il coinvolgimento è totale, il cliente partecipa persino alle riunioni settimanali dei programmatori; in altri casi, il

cliente è coinvolto in una prima fase di progettazione e poi non più; in altri ancora il cliente partecipa indirettamente e viene usato come test della versione rilasciata;

- *Progettazione e documentazione* - Pensare che le metodologie leggere eliminino la progettazione e la documentazione è un errore, in effetti non è così, le metodologie leggere introducono un'iterazione nel ciclo di vita del progetto; quanta progettazione fare e quanta documentazione produrre, escludendo i casi estremi, è una scelta lasciata a chi gestisce il progetto e spesso i teorici dell'Agile Alliance avvisano che è un errore trascurare o addirittura omettere queste due fasi;
- *Consegne frequenti* - Effettuare rilasci frequenti di versioni intermedie del software permette di ottenere più risultati contemporaneamente: si ricomincia l'iterazione avendo già a disposizione un blocco di codice funzionante in tutti i suoi aspetti, si offre al cliente "qualcosa con cui lavorare" e lo si distrae così da eventuali ritardi nella consegna del progetto completo, si usa il cliente come se fosse un test visto che utilizzerà il software e riscontrerà eventuali anomalie, si ottengono dal cliente informazioni più precise sui requisiti che probabilmente non sarebbe riuscito ad esprimere senza avere a disposizione utilità e carenze del progetto;
- *Gerarchia* - La scelta di creare una struttura gerarchica all'interno del team di sviluppo dipende molto dall'approccio del project manager, in ogni caso si ha una conseguenza non secondaria facendo questa scelta; se si decide per una struttura gerarchica ad albero e frammentata si ottiene la possibilità di gestire un numero molto alto di programmatori e di lavorare a diversi aspetti del progetto parallelamente; se si decide per una totale assenza di gerarchia si avrà un team di sviluppo molto compatto e motivato, ma necessariamente piccolo in termini di numero di programmatori;
- *Miglioramento della conoscenza* - Nata con l'avvento della programmazione Object-Oriented, non è altro che la presa di coscienza della produzione di conoscenza che si fa in un'azienda man mano che si produce codice; questa conoscenza prodotta non deve andare perduta ed è per far ciò che si sfruttano

spesso le altre pratiche, come la comunicazione stretta o la condivisione della proprietà del codice;

- *Retroingegneria* - Ossia ottenere, spesso in maniera automatica, la documentazione a partire dal codice già prodotto; è una delle pratiche più diffuse e più controverse, diffusa perché permette un guadagno enorme in termini di tempo, ma controversa perché spesso la documentazione prodotta è inutilizzabile oppure è prodotta solo per una richiesta burocratica del cliente e non verrà mai realmente utilizzata;
- *Semplicità* - Uno dei punti chiave delle metodologie leggere, direttamente mutuato dalla programmazione Object-Oriented, è la semplicità; semplicità nel codice, semplicità nella documentazione, semplicità nella progettazione, semplicità nella modellazione; i risultati così ottenuti sono una migliore leggibilità dell'intero progetto ed una conseguente facilitazione nelle fasi di correzione e modifica;
- *Formazione di una squadra e Proprietà del codice* - La formazione del team di sviluppo è condizionata dalla scelta sulla gerarchia interna, ma segue regole precise che permettono di ottenere un team produttivo nell'ambito della metodologia scelta; la scelta dei membri del team è condizionata anche alla scelta della proprietà del codice, che può essere individuale o collettiva; nel primo caso la responsabilità sullo sviluppo è individuale, nel secondo dipende da tutto il team e quindi dal project manager;
- *Test* - Pratica diffusissima anche prima della nascita delle metodologie leggere, ha prodotto una letteratura vastissima ed una serie di approcci differenti come il Rapid Testing o il Pair Testing; nell'ambito delle metodologie leggere vengono spesso utilizzati insieme tre tipi di test differenti: i test funzionali, utilizzati per verificare che il software faccia effettivamente ciò che è previsto debba fare, i test unitari, utilizzati per verificare che ogni pezzo di codice funzioni correttamente, e i test indiretti effettuati inconsciamente dal cliente ogni volta che gli si consegna una versione;
- *Controllo della versione* - Una delle conseguenze dirette dell'iterazione nella produzione è la necessità di introdurre un modello, un metodo, uno strumento,

per il controllo delle versioni del software prodotto e rilasciato; uno degli strumenti più diffusi e maggiormente suggeriti per ottemperare automaticamente a questa pratica è il CVS.

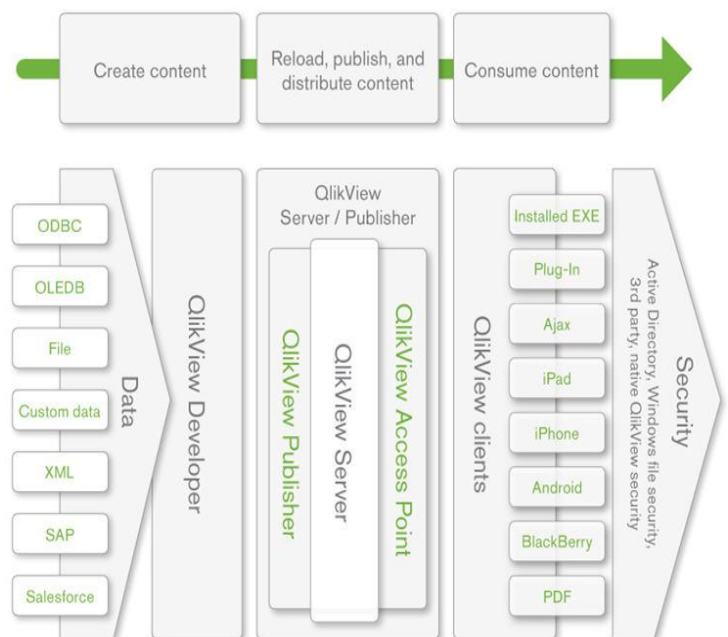
# Capitolo 3. Lo strumento di sviluppo: QlikView

## 3.1 QlikView in breve

QlikView è una piattaforma di Business Intelligence realizzata da QlikTech che permette l'analisi di dati a differenti livelli di aggregazione, tramite la creazione, la modifica e la pubblicazione di report, cruscotti direzionali o altri strumenti utili per l'analisi dei dati. Si basa sul presupposto che "l'analisi dovrebbe agire come la mente umana", quindi con il suo funzionamento cerca di approssimare quanto più possibile il ragionamento dell'utente durante il processo di analisi dei dati. È uno strumento altamente intuitivo che non necessita di particolari conoscenze di modellazione dei dati.

### 3.1.1 Struttura di QlikView

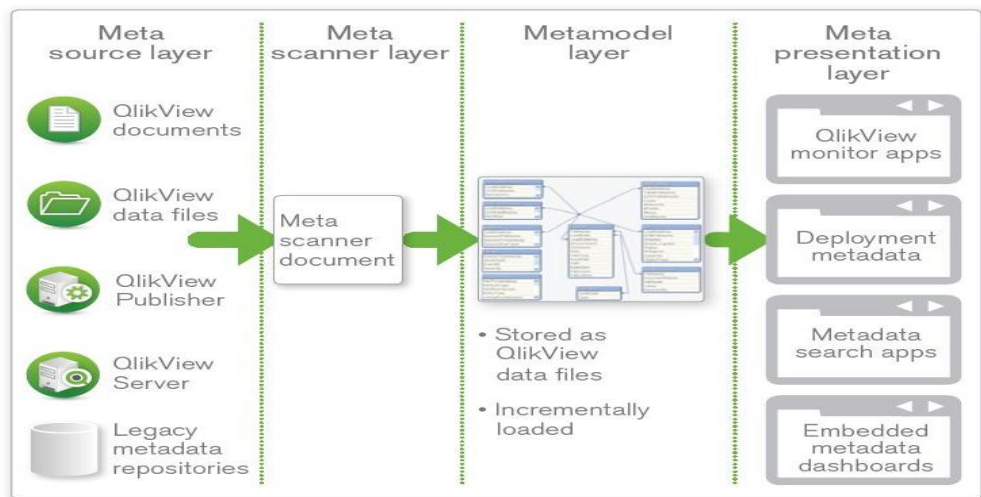
- **QlikView Server**
  - Gestione e apertura dei report
  - Aggiornamento dei report e schedulazione
  - Profilazione utente
- **QlikView Publisher**
  - Esportazione automatizzata report in diversi formati
  - Invio automatizzato report via mail
- **QlikView Access Point**
  - Portale di pubblicazione e consultazione dei report
- **QlikView Desktop**
  - Creazione e modifica dei report



### 3.1.2 L'architettura di QlikView

È in grado di leggere da differenti sorgenti dati (database, files, ecc.) e adotta una logica di tipo associativo, pertanto effettua i join tra le tabelle sulla base della nomenclatura delle colonne (questo può diventare un problema quando tabelle che non sono in relazione hanno delle colonne con nomi uguali). Utilizza una tecnica di analisi

#### Architectural view of QlikView metadata model



© 2011 QlikTech

in-memory, ovvero carica tutti i dati in memoria ed esegue le aggregazioni ed i join in RAM, man mano che l'utente effettua le selezioni.

Questo contribuisce a fornire all'utente un'interfaccia veloce, intuitiva e di semplice utilizzo. Utilizzata in questo modo, l'associazione è il concetto che rispecchia più da vicino il percorso della mente umana – un percorso che non segue un criterio lineare, ma si muove per associazioni. Sfruttando al massimo la RAM disponibile può portare ad un rallentamento delle altre applicazioni che stanno girando sulla macchina. In realtà l'utilizzo di RAM non è costante ma subisce un incremento durante l'importazione dei dati ed il caricamento di questi nei report realizzati, poiché è in quel momento che realizza le associazioni. Una volta che i dati sono caricati, la loro analisi richiede una quantità di memoria costante a meno di piccole variazioni dovute alle operazioni di filtraggio. Quindi lo svantaggio della RAM occupata viene compensato da una maggiore velocità nella disponibilità dei dati e nella loro esplorazione. Considerando la legge di Moore, questo non sembra un problema così vincolante. Ogni nuova generazione di piattaforme di processori consente a QlikView di gestire

una quantità di dati sempre maggiore. Grazie alla disponibilità di sistemi multi-core a 64 bit, le possibilità offerte da QlikView sono oggi virtualmente illimitate.

Alla base di ogni report Qlikview vi è uno script di caricamento dei dati scritto in AQL (Associative Query Language) un linguaggio proprio dello strumento e brevettato dalla QlikTech.

Nella terminologia usata da QlikTech gli oggetti (file) prodotti con QlikView vengono chiamati "documenti". L'estensione di tali documenti è "\*.QVW" e la loro dimensione in relazione ai dati contenuti è molto compressa. Ciascun documento rappresenta una "vista dinamica" sulla base dati e contiene:

Le regole di accesso e caricamento dei dati: tramite connessione ODBC o OLE-DB, integrazione di dati XML, di file di testo o di dati presenti su file MS-Excel

I metodi di compressione da utilizzarsi

I collegamenti tra le varie basi dati (Join).

Le selezioni o parzializzazioni da effettuarsi sui dati stessi, a livello di campi o di record

Le istruzioni e le funzioni di elaborazione, trattamento, aggregazione, calcolo, formattazione ed eventualmente di generazione dei dati

La security, ovvero il controllo degli accessi al documento e le regole di personalizzazione dei dati in base all'utente

I layout di presentazione dei dati e gli oggetti grafici che consentono all'utente di navigare ed analizzare dinamicamente i dati

Le istruzioni per l'integrazione con altri applicativi dell'utente e per l'automazione di particolari funzioni .

Quindi un documento QlikView risulta essere completo ed autosufficiente in modo da consentire all'utente l'analisi dei dati stand-alone (localmente alla macchina) o da remoto. Un documento può essere composto da più schede e in ognuna è possibile inserire diversi elementi, come grafici, tabelle semplici e tabelle con campi calcolati (ad esempio percentuali, somme, differenze, ecc.). In figura 2.2.1 è apprezzabile uno screenshot dell'interfaccia grafica. Come si può notare ciascun tab ed i singoli

componenti sono altamente personalizzabili, permettendo quindi all'utente di applicare all'ambiente il layout secondo il proprio gusto.

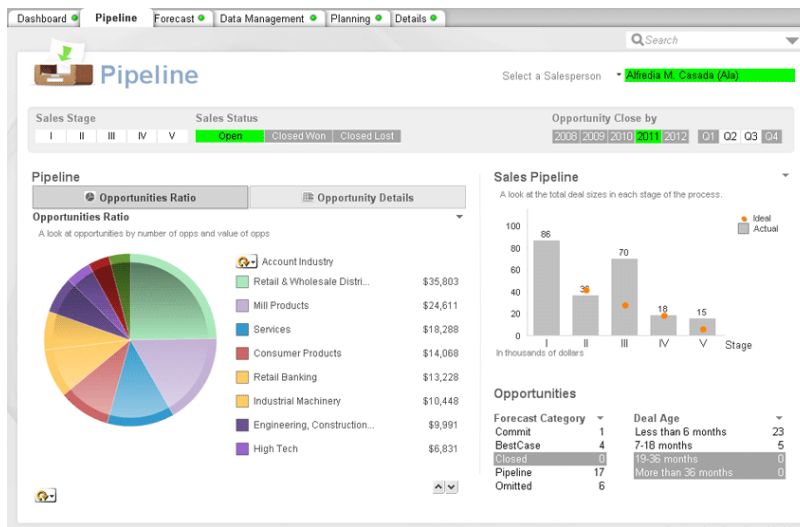


Figura 2.2.1 – QlikView: Interfaccia grafica

I principali componenti da utilizzare per l'analisi dei dati sono list boxes, pivot table, combo chart, line chart, pie chart e scatter chart. In figura 2.2.2 è presente un esempio per alcune tipologie di elementi elencati, ma ve ne sono molti altri ancora.

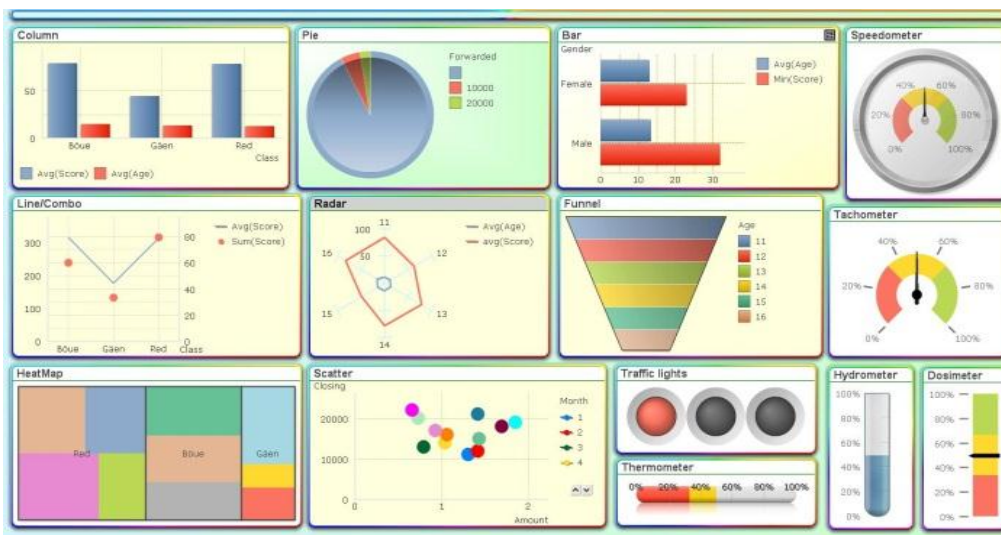


Figura 2.2.2 – QlikView: componenti grafici

Tutti i valori presenti all'interno dei componenti sono selezionabili, anche quando sono rappresentati in maniera grafica, ed ogni singola vista sui dati può essere il punto



di partenza di un'analisi e di una query. E' estremamente facile presentare i dati sia in maniera sintetica, con l'uso ad esempio di intuitivi cruscotti, sia con il massimo dettaglio, lasciando all'utente la possibilità di procedere con un'esplorazione in drill-down, qualora richiesto. Tramite semplici funzioni è possibile creare sofisticate simulazioni ed analisi 'what if?' a partire dai dati disponibili.

Realizzati i report o dei filtraggi su questi, QlikView garantisce il massimo grado di esportabilità dei dati, prevedendo diversi formati di esportazione:

- *MS Excel (.xls/xlsx)*: esportazione dei dati oggetto della selezione su di un foglio di MS Excel. Oltre ai dati, viene anche inclusa una nota dettagliata con l'elenco dei valori delle selezioni effettuate per ottenere tali dati. Nel caso di una esportazione globale la nota sarà 'No selections'.
- *File di testo (.txt)*: i dati vengono inseriti in un file di testo delimitato da virgola, punto e virgola o da tabulazione
- *File di ipertesto (.html, .htm)*
- *File Xml*
- *Immagine in formato .pgm , .jpg , .jpeg o .bmp.*

Per i formati di esportazione sono previste le codifiche ANSI, UNICODE e UTF-8.

## **Set Analysis**

Nel seguente progetto, è stato fatto largo uso della Set Analysis.

La Set Analysis garantisce una forte ottimizzazione nelle logiche di calcolo del SW, riducendo notevolmente i tempi nelle operazioni calcolate a Runtime.

1. È concettualmente simile a una selezione.
2. Fornisce un metodo per definire insiemi (set) di informazioni indipendenti dalla selezione corrente.
3. È possibile creare insiemi di dati basati su altri insiemi.
4. Deve essere utilizzata in una funzione di aggregazione.
5. È sempre contenuta in parentesi graffe { }

Può eliminare la necessità di un'ulteriore complessa codificazione all'interno dell'applicazione.

- Non c'è bisogno di isole dati (anche se le isole dati a volte risultano utili se usate assieme ai set di dati)
- Riduce la necessità di usare macro

Offre una maggiore flessibilità nell'analisi

Si possono aggiungere espressioni per i dati al di fuori della selezione Corrente.

## 3.2 QlikView Best practices

Dopo aver seguito diversi corsi di formazione con QlikView ho avuto la possibilità di apprendere il modo migliore di sviluppare con questo software. Le regole di best practices descritte qui di seguito andranno a rappresentare uno standard di fatto per la progettazione con questo Sw in Azienda.

### 3.2.1 Segmentazione delle cartelle di progetto

Per una corretta gestione e standardizzazione dei progetti, adopereremo la seguente suddivisione.

Folders:

1. DataSource
2. Documentation
3. DW\_Staging
4. DW\_Associative
5. Setting
6. Qvw\_ETL
7. Qvw\_Presentation

#### 1. DataSource

In questa cartella andiamo ad inserire tutti i file che ci permettono di gestire le info esterne: vedi file excel, sql, csv, o altre fonti.

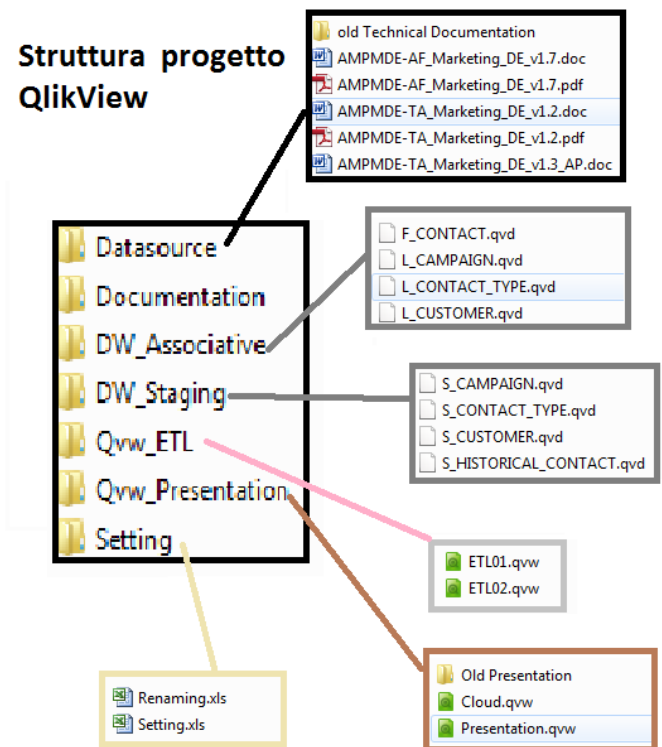
#### 2. Documentation

In questa cartella inseriamo tutta la documentazione utile ai fini del progetto.

E' buona norma tracciare le evoluzioni delle documentazioni attraverso dei versionamenti incrementali.

#### 3. DW\_Staging

Questa cartella conterrà tutti i file QVD (Qlik View Data) del primo livello: ETL01 (Extraction Step).



In altre parole nella fase di ETL01 importeremo tutte le tabelle che ci serviranno all'interno di questa cartella, trasformate in formato qvd (formato QlikView proprietario che permette di risparmiare spazio e migliora le prestazioni).

Questo permetterà una lettura delle tabelle per la fase di Staging, in tempi notevolmente ridotti, rispetto alle letture sulle sorgenti.

#### **4. *DW\_Associative***

Questa cartella conterrà tutti i file QVD (Qlik View Data) del secondo livello.

In altre parole nella fase di ETL02 andremo a costruire o perfezionare la logica relativa alla Staging Area, attraverso elaborazioni delle tabelle importate, ed andremo ad inserire le nuove tabelle elaborate in questa cartella.

Le nuove tabelle create serviranno ad alimentare il file Cloud.qvw che andrà a costruire la Nuvola (equivalente al DataMart in Qlikview).

#### **5. *Setting***

Questa cartella conterrà dei file di configurazione.

1) *Setting.xls* servirà a controllare le stringhe di connessione o di estrazione e caricamento delle varie fasi di ETL/Cloud in forma centralizzata:

- Tab\_1: parametri di connessione centralizzati,
- Tab\_2: variabili utilizzate (Opzionali),
- Tab\_3: tabelle da estrarre (Contiene la lista delle tabelle che si vogliono caricare nel sistema, dalle diverse sorgenti).

2) *Renaming.xls* , servirà a definire la fase di rinomina delle Dimensioni/Fact Table e dei loro Metadati, in maniera controllata e centralizzata, per le diverse fasi di elaborazione (es. ETL02 e Cloud)

*Renaming.xls* è così composto:

- Tab\_1: *Renaming\_Tables* (contiene la lista centralizzata dei nomi che si vuole assegnare alle singole tabelle nelle diverse fasi),
- Tab\_2: *Renaming\_Metadata* (contiene la lista centralizzata dei nomi che si vuole assegnare ai campi delle tabelle nelle diverse fasi).

Nel caso l'utente decida di effettuare variazioni ai nomi dei campi o delle tabelle che sono state importate, o si decida di importare nuove tabelle, operando su questi file si può gestire in maniera centralizzata il renaming.

## **6. *Qvw\_ETL***

Questa cartella conterrà tutti i file di ETL, con le varie logiche di primo livello e secondo livello.

Le principali operazioni effettuate in questi file sono:

- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization

I file potrebbero essere anche suddivisi se i progetti si rilevano lunghi e complessi.

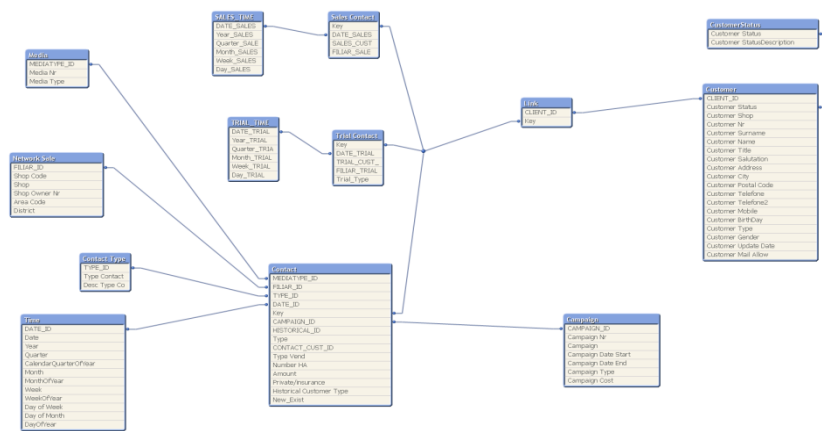
Inoltre, suddividendo i file, si possono effettuare operazioni parallelizzate del flusso, permettendo così una maggior rapidità nei task di caricamento e costruzione della logica.

- 1) *ETL01*: in questo file verrà generato il codice che conterrà la prima fase di Estrazione-Trasformazione-Caricamento. Il file viene costruito in modo da centralizzare tutte le operazioni di lettura dei file dalle sorgenti e cercando di limitare, ove possibile, le operazioni di elaborazione sui file sorgente (più lente rispetto alle operazioni sui file qvd). Alla fine del processo di estrazione verranno salvate tutte le tabelle in formato qvd, nella cartella *Dw\_Staging*.
- 2) *ETL02*: in questo file andremo a leggere le tabelle da *Dw\_Staging*, completeremo le operazioni di Staging, costruzione del modello multidimensionale, ed infine, prima di salvare i nuovi qvd in *Dw\_Associative*, andremo ad effettuare la prima operazione di renaming delle tabelle e dei metadati.

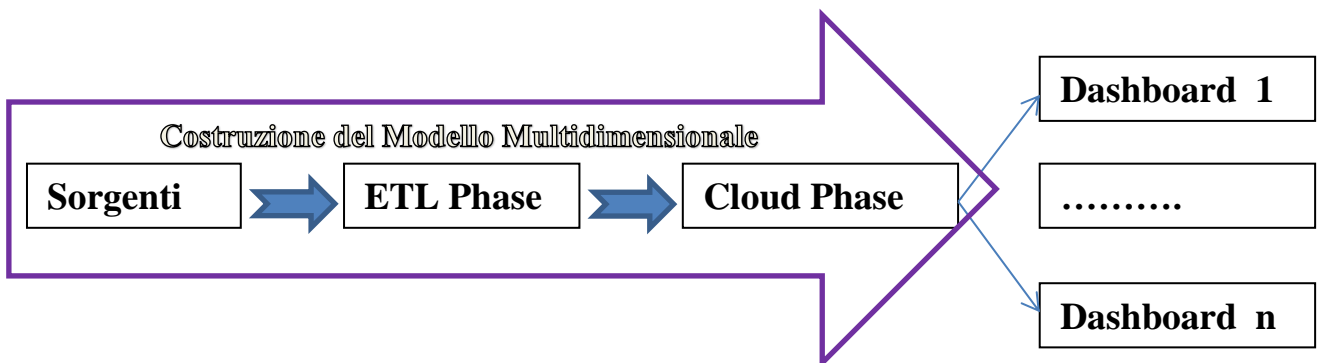
## **7. *Qvw\_Presentation***

In questa cartella andremo a suddividere la fase di costruzione del Datamart (Cloud phase) e di generazione dei report (Presentation phase).

- 1) *Nuvola.qvw*: conterrà la logica di costruzione della nuvola (che corrisponde al Datamart). Per costruirlo, il progettista dovrà riferirsi al manuale per capire le logiche di Join imposte da Qlik View (es. utilizzare gli stessi nomi per generare un join fra tabelle). Per una maggior automazione del processo si è scelto di svincolare il processo di costruzione della Nuvola (Tab: Costruzione\_Nuvola) dalla fase di Renaming dei nomi (Tab: Renaming).

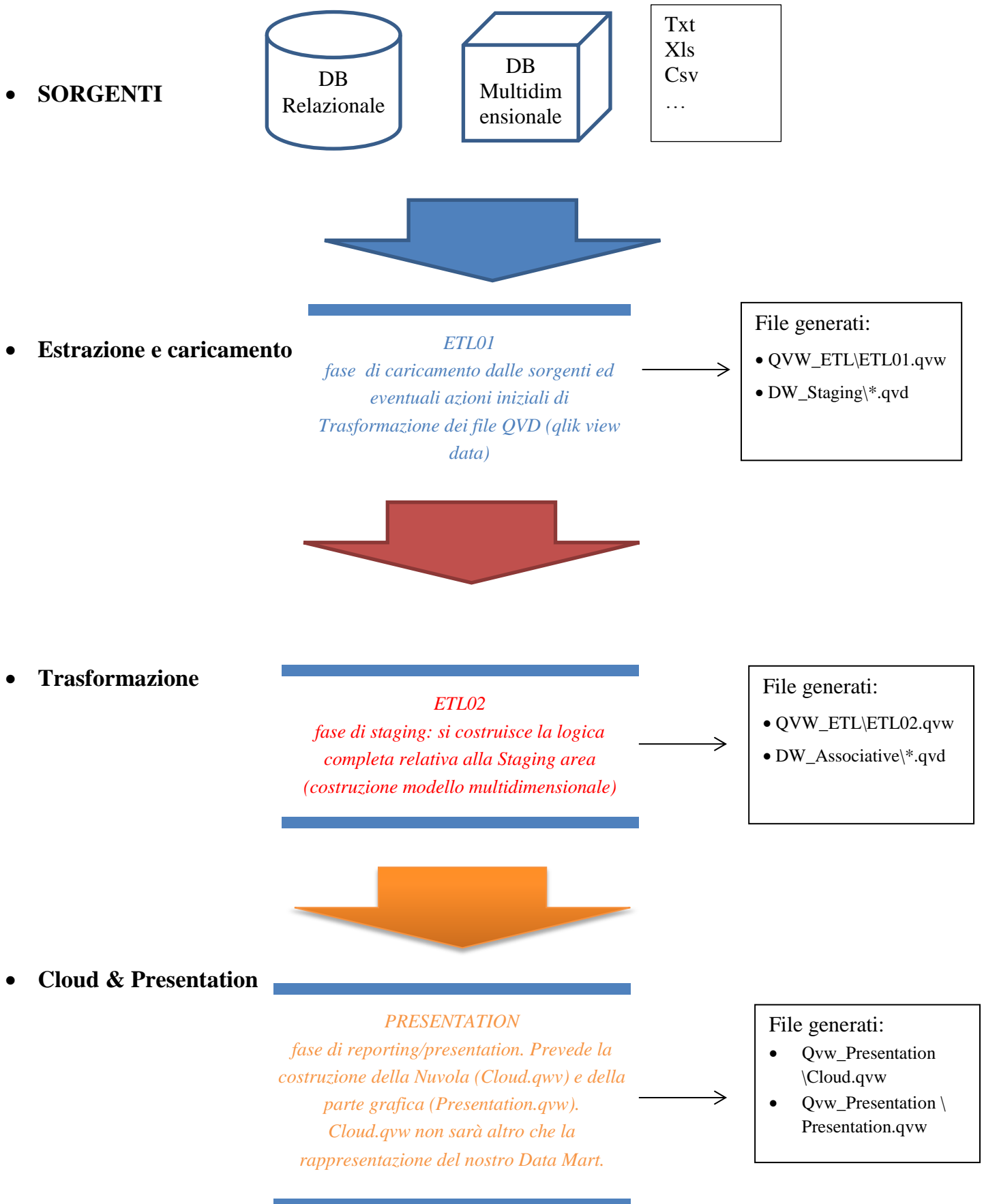


- 2) *Presentation.qvw*: sarà, invece, file privo di script (semplicemente includerà *Nuvola.qvw*), dove i progettisti/designer si potranno dedicare alla sola fase di costruzione delle Dashboard e dei Report per le analisi.



Una corretta progettazione permetterà la facile generazione di flussi paralleli che offriranno la possibilità di costruire Presentation diversi, ma che condividono lo stesso modello (es. riutilizzare il modello costruito per il reparto Marketing, anche per generare i reports del reparto Financial).

### 3.2.2 Flusso base di un progetto





## Capitolo 4. Il caso di studio: Amplifon DE

Nella fase seguente procederemo con la descrizione del caso di studio che è stato preso in considerazione per lo sviluppo delle tecniche e delle logiche progettuali precedentemente descritte.

*Come specificato nella fase introduttiva, è molto importante tenere presente che i dati rappresentati nel corso della tesi (negli screenshot di esempio) saranno offuscati, questo perché l'azienda committente (Amplifon De) non autorizza alla divulgazione in quanto dati sensibili per la stessa.*

Il caso di studio di seguito trattato rappresenta un progetto pilota. È il primo esempio di collaborazione fra l'azienda di consulenza ICONSULTING e il cliente Amplifon. Nello specifico gli attori coinvolti sono stati:

- **Amplifon Germany** (utilizzatore finale e committente del progetto).
- **Amplifon Corporate** (supervisor del progetto)
- **Iconsulting** (sviluppatore del progetto)

Il fine ultimo del progetto è stato quello di realizzare una piattaforma di Business Intelligence in grado di fornire sia un'efficace che efficiente sistema di analisi e supporto alle decisioni per il reparto Marketing di Amplifon DE. La richiesta, fatta da Amplifon Germany, ma supportata da Amplifon Corporate, ha permesso di sviluppare un'efficace sistema di analisi delle campagne promozionali, in funzione di fattori strategici, come nuovi clienti generati, efficienza o inefficienza promozionale, ecc.



## 4.1 Descrizione Funzionale

Obiettivo di questa fase è quello di descrivere il modello di implementazione delle relazioni richieste nell'ambito del progetto "Amplifon DE marketing Report".

Il set di strumenti descritti nei paragrafi seguenti è un'applicazione personalizzata, finalizzata a fornire un'analisi a sostegno del reparto Marketing di Amplifon DE. In particolare il sistema ha lo scopo di evidenziare: l'efficacia delle campagne promozionali, la capacità di conversione dei Test in Trial e delle Trial in Sales, conteggiare i test dell'udito, ecc. le norme specifiche, che si applicano al contesto tedesco, saranno attuate nei KPIs<sup>1</sup> successivamente descritti.

Descriveremo nel dettaglio le esigenze degli utenti emerse dopo diverse riunioni e scambio di email con i reparti IT, commerciale e di marketing.

### 4.1.1 Scopo del progetto

L'intento del progetto è stato quello di costruire un nuovo sistema di reportistica basato su tecnologia QlikView. Nel processo attuale, il dipartimento Marketing di Amplifon Germany effettua analisi basate su report costruiti ad hoc dal reparto IT locale e basate sul sistema di CRM aziendale.

Il nuovo strumento ha lo scopo di diventare l'unico punto di verità in termini di analisi delle prestazioni e reporting per il reparto marketing tedesco, che consente all'organizzazione di fornire informazioni e analisi integrate e certificate.

Il nuovo ambiente di informazione offrirà dati elaborati e provenienti dal SIM\_DE<sup>2</sup> (alimentato da GOAL<sup>3</sup>) al fine di realizzare un sistema orientato al reporting. Questa elaborazione (processo ETL) sarà eseguita in QV.

I KPI richiesti dal Dipartimento di Marketing tedesco sono descritti in dettaglio più avanti.

Il tipo di analisi richiesta da Amplifon De può essere raggruppata in cinque report flessibili, fornendo allo stesso tempo diversi tipi di analisi, attraverso le funzionalità di navigazione offerte da QlikView.

---

<sup>1</sup> Key Performance Indicators: è un indice che monitora l'andamento di un processo aziendale.

<sup>2</sup> Sistema di storicizzazione dei dati tedesco che si appoggia su GOAL.

<sup>3</sup> Sistema Legacy usato per centralizzare le informazioni per la maggior parte delle Country (DB relazionale)

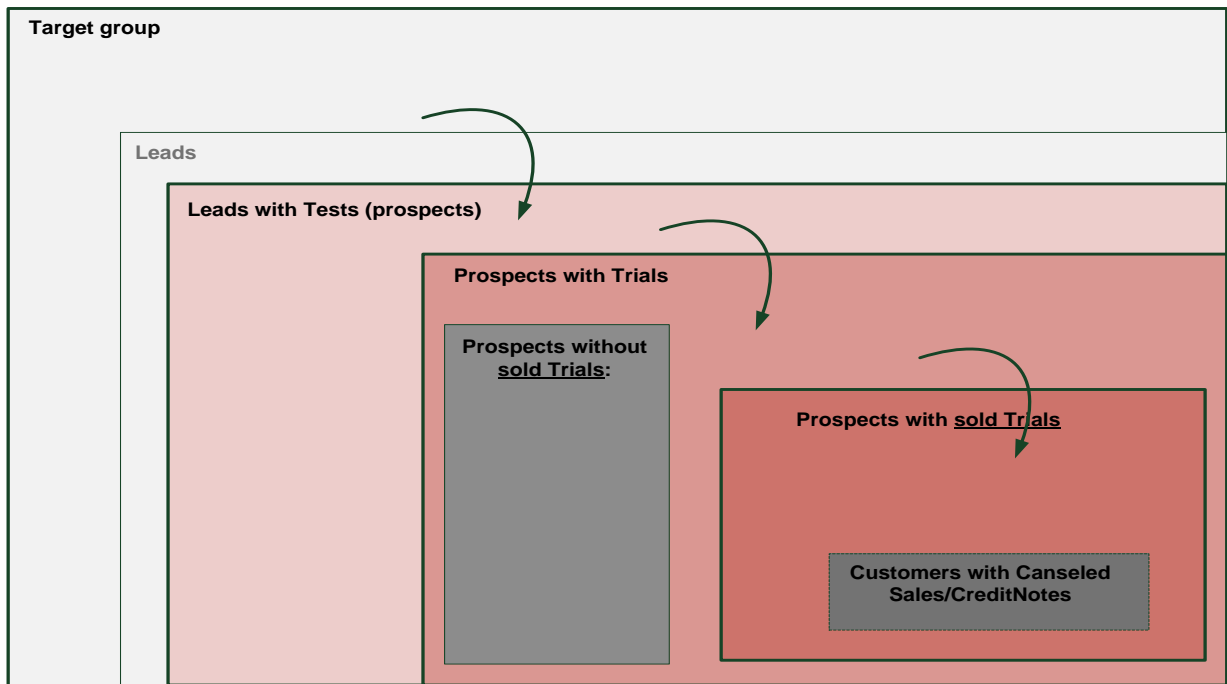
La lista dei report sviluppati è la seguente:

- **Campaigns Reporting Area.** Essa copre le seguenti aree di analisi:
  - Risultati generali per Campagna;
  - Risultati dei Negozi per Campagna;Per entrambi è garantita un'analisi addizionale, basata sulla possibilità di suddividere per il tipo di Campagna promozionale ( es. “advertising campaign”, “mailing campaign”, “national campaign”).
- **Media Type** offre le seguenti analisi:
  - Analisi dei contatti in funzione dei media a cui sono riferiti;
- **Funnel and Conversion Rates Reporting Area** offre le seguenti analisi:
  - Conversions rates Test -> Trials, Trials -> Sales
  - Sales funnel;
- **Customer Hering Aids Reporting** offre le seguenti analisi:
  - Anagrafica dei Clienti che hanno avuto almeno un contatto per negozio;
- **Hearing Tests Analysis Reporting Area** offre le seguenti analisi:
  - Conteggio degli Hearing Test.

#### 4.1.2 Il processo di business

##### Variazione dello Stato

Lo stato di un cliente può variare in base a diversi eventi che possono verificarsi (vedi figura sotto)



## Terminologia

- *Lead (LD)*: cliente che ha ottenuto solo l'iscrizione nel sistema di front office. Un lead può essere identificato anche come colui che non sia una delle seguenti tre categorie.
- *Prospect (PC)*: Lead che ha effettuato un test dell'udito;
- *Client (CL)*: cliente che ha acquistato almeno una HA;
- *Competitive User (CU)*: cliente che in precedenza ha acquistato un HA da un concorrente;
- *Existing Client*: al momento di un evento registrato (Test, Trial o Sales), un cliente che era già cliente di Amplifon (Tipo storico = "CL");
- *New Client*: al momento di un evento registrato (test, le prove o di vendita), colui che non era ancora un cliente di Amplifon (non ha mai acquistato un Hearing Aids).

## Stati Possibili

Una persona che viene registrata nel sistema può assumere diversi stati possibili:

- AX - AXA Insurance

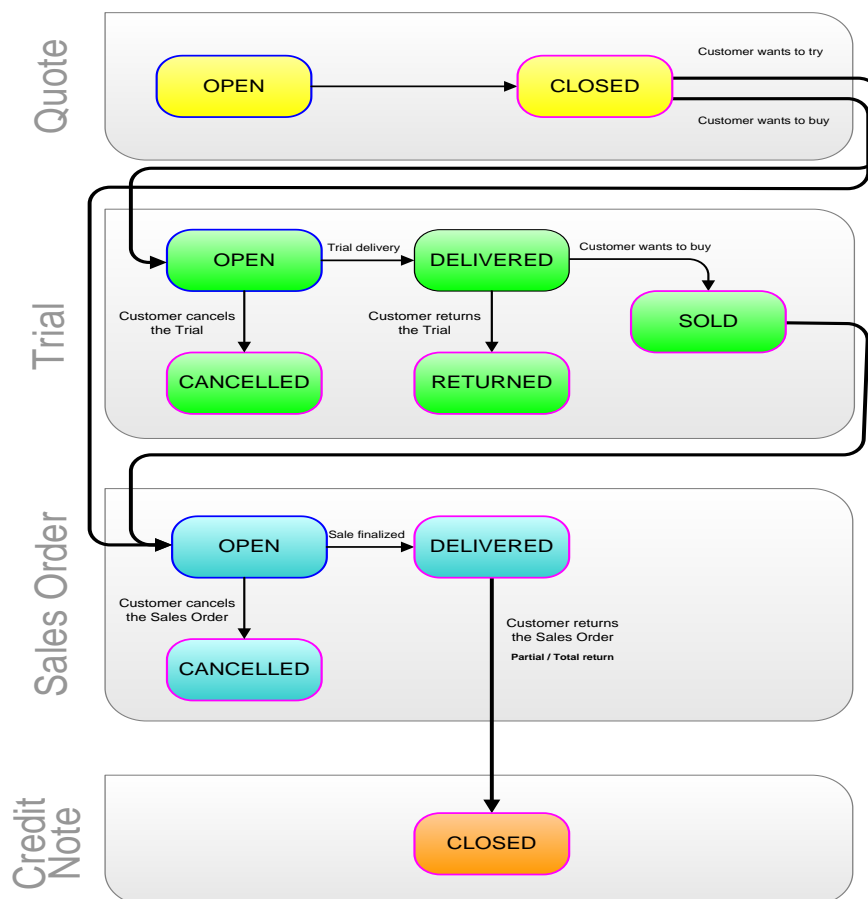
- CR - Credit notes
- DS - Quick sales
- EW - Extended Warranty
- LD - loss and Damage
- QU - Quote
- RE - Repair
- RR - Registration
- SA - Sale
- SZ - Historic data
- TR – Trial.

Verranno caricati tutti gli stati possibili anche se, ai fini dell'analisi, risulteranno strategici un sottoinsieme di questi:

- ✓ Registration: evento che rappresenta la registrazione di un soggetto al sistema;
- ✓ Test: evento relativo alla prova di test effettuata dal soggetto;
- ✓ Trial: evento relativo alla prova del dispositivo acustico effettuato dal soggetto;
- ✓ Sales: evento relativo all'acquisto del dispositivo acustico;
- ✓ Credit Note: evento relativo alla resa di un dispositivo precedentemente acquistato.

Gli altri verranno caricati per dare la possibilità di effettuare analisi generiche sul cliente, ma non saranno strategici.

Nel seguente grafico si nota chiaramente il processo tipico evolutivo che si vuole tracciare.



*Durante il primo sviluppo è stata stimata l'attività relativa ad una vendita come semplice contatto di tipo Trial Sold.*

In seguito, a causa di valutazioni errate da parte dell'organismo di supporto (Amplifon Corporate), sono state variate le logiche del funnel considerando gli eventi di tipo Sales (più reali) e non eventi di tipo Trial Sold (che non sono pienamente significativi, a causa di inconsistenze dei dati nel sistema di riferimento GOAL). Queste variazioni sono state gestite come secondo progetto detto CR (Change Requirement), come evoluzione del primo, in quanto prevedeva sostanziali modifiche al modello.

### **Il processo delle Trial**

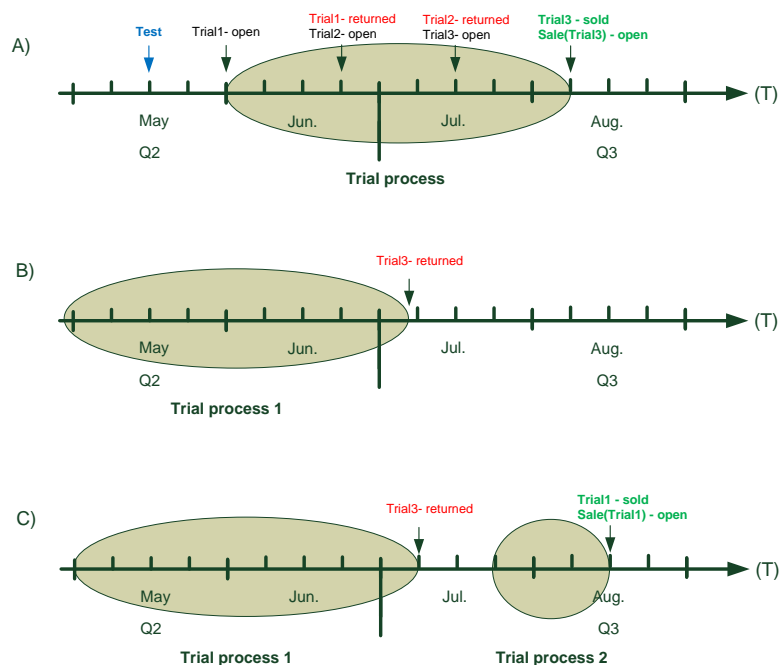
In Amplifon Germany più di una trial verrà considerata, (in media 2) per un *Trial Period*, così da rendere possibile una corretta analisi delle performance di vendita basate sulla semplice conta dei documenti di vendita nel sistema di *Front-end* (la componente che interagisce col cliente). Questa risulta una normale situazione, in quanto non tutte le trial si convertono immediatamente in vendite effettive. Per ogni

*Trial Process* vengono considerate diverse trials e l'intero processo dovrà finire con una vendita, per essere considerato tale. Generalmente possono presentarsi tre diversi scenari per il processo di trial: A,B e C (vedi schema sotto).

A) E' possibile che nel corso di una procedura si verificano diverse trials. Tutte le trials non hanno alcuna relazione con le altre ad eccezione di due parametri: il periodo di tempo comune e il numero cliente. Questo rende impossibile un'analisi delle trials relative a un Trial Process (ad esempio il conteggio delle trial) senza definire un periodo di tempo tra tutti loro. Solo le vendite effettuate possono riferirsi alla prova precedente. Le vendite sono rappresentate dallo stato Sold della Trial.

B) E' possibile che un Trial Process non abbia un stato Trial Sold e così termini senza che ciò comporti alcuna vendita.

C) È possibile che un Trial Process possa essere diviso in due parti, con ritardo fra le trial. In realtà si tratta di un unico processo di prova, ma con tempi diversi di termine ed inizio della Trial successiva. Questo rende impossibile fare l'analisi delle trial in base alle date.



Pertanto, Amplifon suggerisce di utilizzare dei tassi di conversione basati sulla conta dei clienti al posto della conta dei documenti. Questo varrà per ogni stadio dell'analisi: Test, Trial e Sales.

### 4.1.3 Il modello analitico

#### Modello concettuale

Il seguente modello concettuale riassume in breve il cubo, o nuvola per qlikview, che si vorrà rappresentare per rendere visibili e fruibili tutte le funzionalità del sistema richieste dal cliente.

Il modello è stato rappresentato utilizzando una notazione standard detta *DFM* (Dimensional Fact Model).

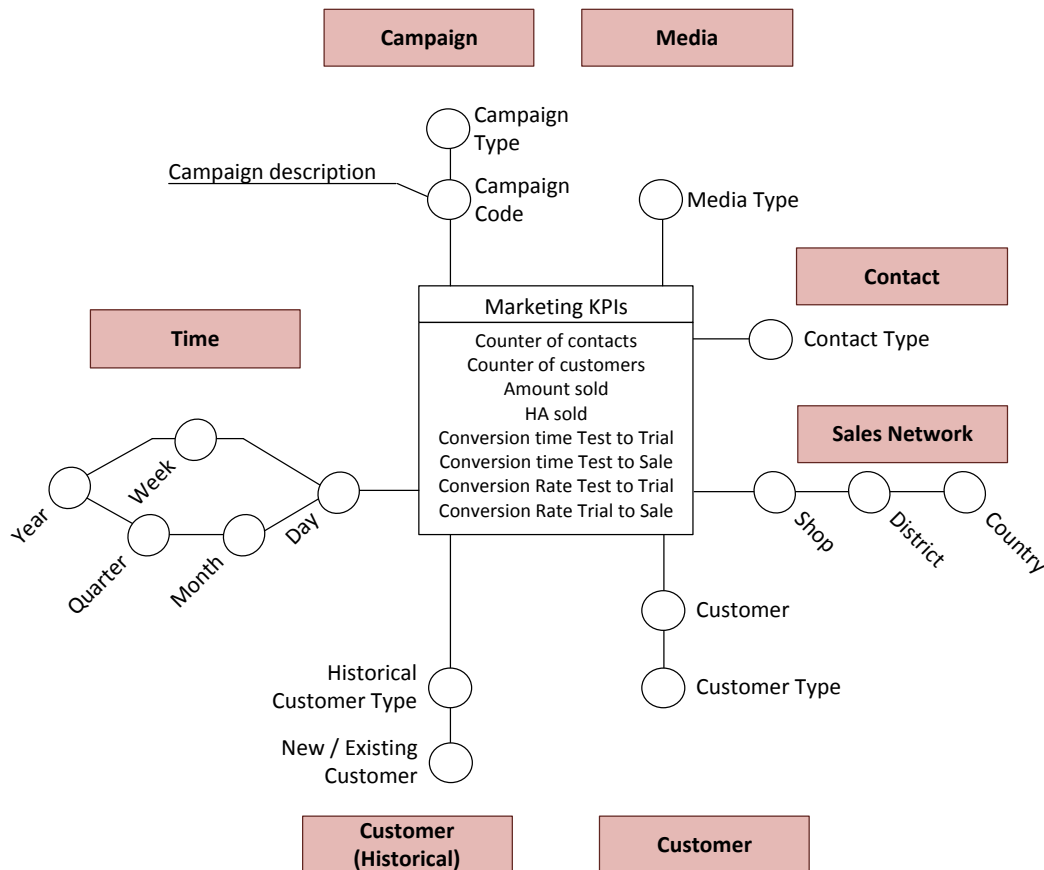
Il DFM è un modello concettuale grafico per data mart, pensato per:

- supportare efficacemente il progetto concettuale;
- creare un ambiente su cui formulare in modo intuitivo le interrogazioni dell'utente;
- permettere il dialogo tra progettista e utente finale per raffinare le specifiche dei requisiti;
- creare una piattaforma stabile da cui partire per il progetto logico (indipendentemente dal modello logico target);
- restituire una documentazione a posteriori espressiva e non ambigua.

La rappresentazione concettuale generata dal DFM consiste in un insieme di schemi di fatto (uno solo nel nostro caso). Gli elementi di base modellati dagli schemi di fatto sono i fatti, le misure, le dimensioni e le gerarchie.

Il modello costruito è uno schema a stella che, come descritto nel capitolo 1, contiene le dimensioni accorpate al livello più basso, e il fatto centrale con le chiavi per le dimensioni e i KPI che rappresentano le misure del business.





La seguente rappresentazione del modello rispecchia una fase primordiale del sistema, successivamente sono stati previsti aggiunti dei campi non originariamente inclusi nel sistema. La *Nuvola* di QlikView, rappresentata successivamente nella fase di Cloud, rappresenterà esattamente il modello completo di tutte le dimensioni e i campi del sistema. Ma la struttura di base del modello resta invariata.

## Le Dimensioni

Le dimensioni, che verranno ora descritte, rappresentano esattamente le dimensioni concordate nella fase iniziale. Successivamente, il modello si arricchirà di eventuali dimensioni e campi di supporto per il raggiungimento della logica e degli obiettivi finali (ad esempio la tabella Customer verrà arricchita di tutta una serie di attributi descrittivi).

Le dimensioni, con i diversi livelli gerarchici sono rappresentate nella tabella sottostante:

<i>Dimension</i>	<i>Level</i>	
<b>Time</b>	4	Year
	3	Quarter
	2	Month
	1	Day
<b>Campaign</b>	2	Campaign Type
	1	Campaing (code and desc.)
<b>Sales Network</b>	3	Country
	2	District
	1	Shop
<b>Customer</b>	2	Customer Type
	1	Customer
<b>Customer (Historical)</b>	2	New Customer / Existing Customer
	1	Historical Customer Type
<b>Media</b>	1	Media Type
<b>Contact</b>	1	Contact Type

Il nuovo strumento adotterà un approccio lineare per le gerarchie. Ciò significa che i reports si adatteranno automaticamente all'ultima versione di una gerarchia e di conseguenza anche le informazioni. Per esempio, se un negozio si sposta da un quartiere all'altro, le vendite di questo negozio sia in corso, sia dell'anno precedente, saranno conteggiate come parte del nuovo quartiere.

## **I KPIs**

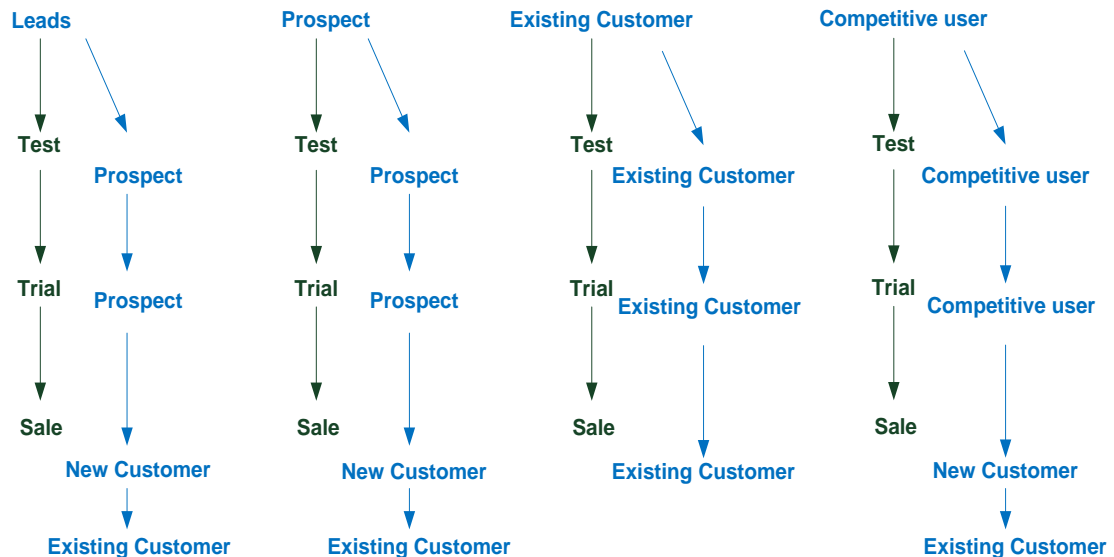
Adesso procederemo nella descrizione dei KPI richiesti (le misure). La descrizione seguente riguarda un aspetto esclusivamente funzionale.

Tre KPI di base vengono utilizzati per il calcolo di tutti gli altri KPI.

- Count of Clients: conteggio dei clienti distinti.
- Count of Contacts: conteggio dei contatti distinti.

- HA sold: numero di unità Hearing Aid (Dispositivi acustici) vendute.

Nel grafico sottostante, invece, vedremo la logica funzionale relativa ai cambiamenti di stato per un determinato cliente:



*Tipologia degli stati del cliente.*

- Customers (Historical Customer Type = “CL”): colui che ha effettuato un acquisto.
- Competitive Users (Historical Customer Type = “CU”): colui che viene da un’azienda concorrente.
- Prospects (Historical Customer Type = “PC”): colui che effettua un Test o una Trial.
- Leads (Historical Customer Type = “LD”): colui che si è registrato nel sistema ma non ha mai fatto altre operazioni.

*New/Existing.*

- Existing Customer (Historical Customer Type = “CL”): colui che ha effettuato almeno una volta un acquisto di Hearing Aids.
- New Customer (Historical Customer Type = “CU”, “PC” or “LD”): colui che non ha mai fatto un acquisto dal momento della sua registrazione.

Tutti i seguenti KPIs sono legati alla selezione corrente che può essere effettuata per tutte le dimensioni precedentemente descritte.

1. *Count of Clients* (numero di utenti suddivisi per stato storico).
2. *New Generated* (nuovi clienti generati da una campagna ).
3. *Tests* (numero di test effettuati da tutti i clienti).
4. *Trials* (numero di trial effettuate da tutti i clienti).
5. *Sold Trials* (numero di trial sold effettuate da tutti i clienti).
6. *Trials Delivered* (numero di trial delivered effettuate da tutti i clienti).
7. *Tests per Client* (numero di test raggruppati per cliente).
8. *Client with Test* (numero di clienti che hanno effettuato almeno un test).
9. *Clients with Trials* (a. Campaign Report logic: numero di clienti che hanno effettuato almeno una trial; b. Funnel Logic: numero di clienti che hanno effettuato un test ed una trial nel periodo selezionato).
10. *Clients with Sold Trial* (a. Campaign Report logic: numero di clienti che hanno effettuato almeno una trial sold; b. Funnel Logic: numero di clienti che hanno effettuato un test ed una trial Sold nel periodo selezionato).
11. *Conversion Rate Test to Trial* ( $\text{Clients with Trials} / \text{Clients with Tests}$ , valido solo per la logica del funnel).
12. *Conversion Rate Trial to Sale* ( $\text{Clients with Sold Trials} / \text{Clients with Trials}$ , valido solo per la logica del funnel).
13. *Conversion time Test to Trial* (tempo che passa tra il primo test e la prima trial per ogni cliente, calcolato per campagna, considerando il massimo valore e il valore medio dei clienti).
14. *Conversion time Test to Sale* (tempo che passa tra il primo test e la prima trial sold per ogni cliente, calcolato per campagna, considerando il massimo valore e il valore medio dei clienti).
15. *Amount sold* ( Valore netto di vendita, considerando soltanto quelle vendite che non hanno avuto Credite Notes e che appartengono all'ordine dei dispositivi acustici, sottraendo eventuali sconti).

Questi sono principalmente i fattori di business che sono richiesti nel sistema. A questi andremo ad aggiungere operazioni legate alle campagne promozionali, fornendo andamenti e grafici significativi in base al report in cui sono situati.

La tabella seguente mostra una rappresentazione riassuntiva delle interazioni dei principali dei KPI in funzione delle dimensioni e dei campi interessati.

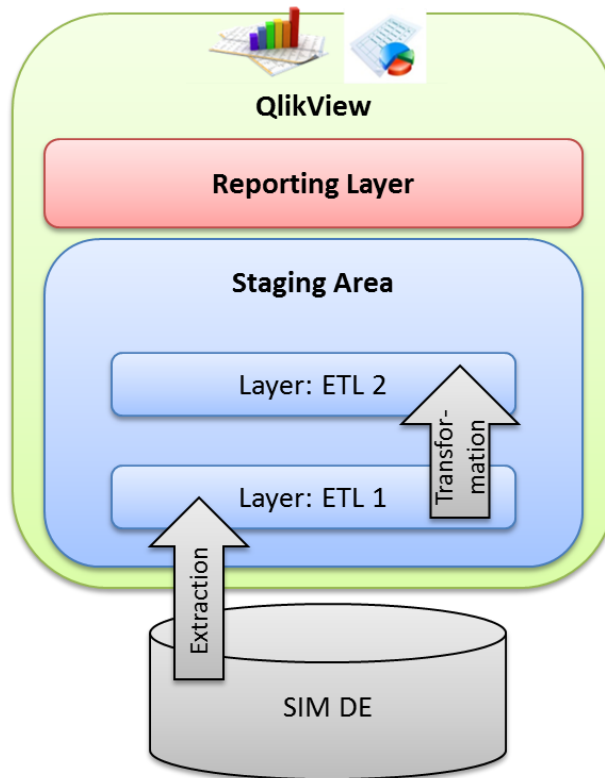
KPIs	Dimensioni	Contact	Customer (Historical)		Customer		Sales Network			Time				Campaign		Media
	Campi	Contact Type	Existing / New	Historical	Customer Type	Customer	Country	District	Shop	Year	Quarter	Month	Week	Campaign Type	Campaign	Media Type
Counter of contacts		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Counter of customers		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Amount sold			X	X	P	X	X	X	X	X	X	X	X	X	X	X
HA sold			X	X	P	X	X	X	X	X	X	X	X	X	X	X
Conversion time Test to Trial			X	X	X	X	X	X	X	X	X	X	X	X	X	X
Conversion time Test to Sale			X	X	X	X	X	X	X	X	X	X	X	X	X	X
Conversion Rate Test to Trial		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Conversion Rate Trial to Sale		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

P = Parziale. Solo per tipi = CL.

#### 4.1.4 Architettura del sistema

L'architettura è stata costruita seguendo esattamente le fasi descritte al paragrafo 3.2. Per la prima volta si è seguita la progettazione di un progetto in QlikView basandosi sul Know How di Best Practices acquisito.

- Tutti i dati provengono dal SIMDE (e non direttamente dal sistema di CRM Aziendale: GOAL).
- La Trasformazione dei dati e il calcolo dei KPIs verranno effettuati nella fase di Staging all'interno dell'ambiente QlikView.
- Basandoci sulle best practices di QlikView, andremo a sviluppare la Staging Area (fase di costruzione del cubo) attraverso le due fasi di ETL (Extraction, Trasformatio and Load), dette ETL01 e ETL02. In queste fasi opereremo con la lettura dei dati dalla sorgente, per poi trasformarli nel modello di business concordato.
- L'ultimo livello di reporting, attraverso il file di Cloud, permetterà l'adattamento della logica di business a quella che è la logica di QV, per poi andare a rappresentare graficamente i Report nel Presentation file.



## L'aggiornamento dell'informazione

*Technomind*<sup>4</sup> ha fornito i dati mancanti nel SIMDE DB e ha permesso la consistenza dei dati dei processi di ETL, garantendo l'aggiornamento del SIMDE ogni notte.

In questo modo un trigger successivo al caricamento del SIMDE verrà schedulato nel QV Server, così da garantire un aggiornamento del sistema giornaliero.

---

<sup>4</sup> società che cura il sistema di CRM (GOAL) per AMPLIFON



## 4.1.5 End User Interface

Di seguito mostreremo le analisi che sono state suddivise su diversi report e che saranno visibili all'interno di una grande dashboard navigabile per reports distinti. Tutte le analisi sono state raggruppate in cinque report principali:

- **Campaign Reporting Area**

Questo report comprende, sulla base dei criteri di selezione dei tipi di campagne come "campagna pubblicitaria", "campagna di mailing", "campagna nazionale", ecc., le seguenti aree di analisi:

- Campaign Type - Risultati generali;
- Campaign Type - I risultati per negozio.

Questo report consente di selezionare e filtrare per il tipo di supporto desiderato: campagne di mailing, campagna pubblicitaria, campagna nazionale, ecc.

Il report è basato su una tabella che può essere navigata in tutte le dimensioni (ad esempio, selezionando, filtrando o facendo drill-down) con le sole limitazioni che sono state descritte nel capitolo 4.2 per il livello massimo di ogni KPI. Ad esempio, la navigazione per la dimensione *Network Sale*, sarà fatta attraversando i livelli di distretto o negozio.

I KPI sviluppati per questo report sono:

- HA Venduti
- Importo venduto
- Prezzo medio (calcolato come: Importo di vendita / HA venduti)
- I clienti con Test (senza logiche di funnel)
- I clienti con Trials (senza logiche di funnel)
- I clienti con Trials vendute (senza logiche di funnel)
- Trial Consegnate
- Ordine di vendita Venduto (senza ritorno o CR)
- I clienti (clienti distinti che hanno acquistato un HA)
- Nuovi contatti generati
- Costi della campagna
- Costo per cliente (Camp. Cost / ordine di vendita venduta)

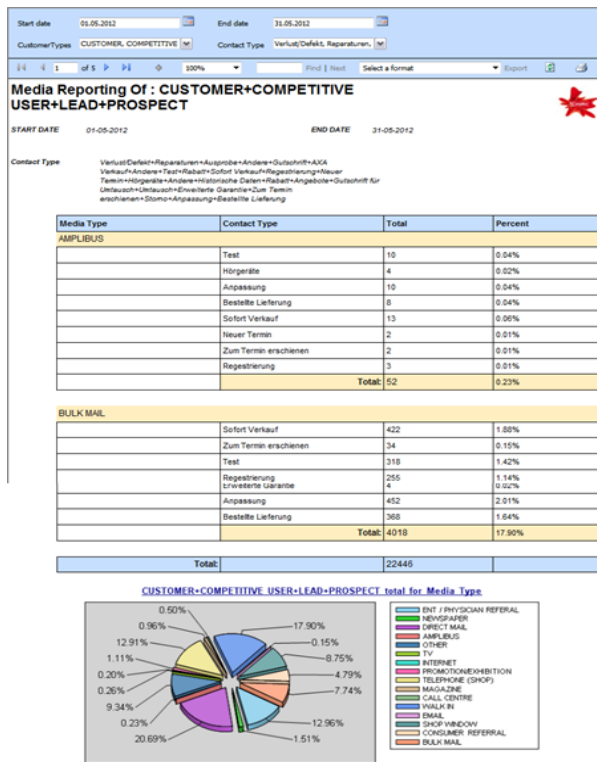
- Costo per Lead (Camp. Cost / Nuovi contatti generati).

Per quanto riguarda le informazioni di vendita, nella presente relazione sono considerate solo le vendite di tipo HA (non altri tipi di vendite).

- **Media Type Reporting Distribution**

Questo report comprende la seguente area di analisi: distribuzione dell'efficacia dei tipi di Campagne promozionali.

Il report permetterà la navigazione su dimensioni di: Time, Campaign Type, Campaign, Customer Type. Inoltre, esso comprenderà sia una tabella, sia un grafico a torta. La tabella può essere navigata sul Contact Type e Media Type, ma il grafico mostrerà sempre le percentuali dei New Generated suddivisi per Media Type.



I KPI di questo report sono:

- Conta dei Clienti distinti che hanno avuto un contatto (nel grafico a torta);
- Conta dei contatti per Media Type(in tabella).

- **Funnel and Conversion Rates**

Questo report comprende le seguenti aree di analisi:

- Conversions rates test-> trial, trial -> sales;
- Sales funnel (per mesi).

Il report include sia tabelle che alcuni grafici. La tabella può essere navigata su tutte le dimensioni (ad esempio selezionando, filtrando o facendo drill-down) con le sole limitazioni sul livello massimo di profondità per ogni KPI.

KPI di questo report sono:

- Conversion Rate Test to Trial
- Conversion Rate Trial to Sale
- Conversion Time Test to Trial
- Conversion Time Test to Sale
- Customers with Tests (Funnel logics)
- Customers with Trials (Funnel logics)
- Customers with Sold Trials (Funnel logics).

Sarà fornita una semplice rappresentazione a imbuto insieme a tre grafici a torta che mostrano la spaccatura tra Existing Customer vs New Customer su Test, Trial e Trial Sold.

- **Customer Hearing Aids**

Questo report comprende la seguente area di analisi: Customer Hearing Aid.

Il report ha poche possibilità di navigazione ed è ottimizzato per essere il più simile possibile all'output richiesto dagli utenti.

L'output desiderato è:

Districts Schleswig-Holstein G1 Shops G1 - Bad Bramstedt, G1 - Bad

years 2012 isOlder  True  False

100% Find | Next Select a format Export

### Customer HA by Shop

DISTRICTS Schleswig-Holstein G1

Shop	Shop Code	Customer Nr	Birthday	Type	Salutation	Title	First Name	Surname	Gender	Street	Postal Code	City	1st last HA Sales Delvry Dt (older than 2012 years)	(1st last) HA private or Insurance	2nd last HA Sales Delvry Dt(older than 2012 years)	(2nd last) HA private or Insurance	3rd last HA Sales Delvry Dt(older than 2012 years)	(3rd last) HA private or Insurance	Author. Provide	Mail Allow
G1 - Bad Bramstedt	844	02400194474520	8/16/1913	CUSTOMER	Herr		Artur	Schwarz	M	Am Hochmoor 107	24539	Neumünster	1/1/1901						N	Y
		02400194474230	1/1/0001	CUSTOMER	Frau		Ilse	Zündel	F	Rugenbusch 122i 112	24576	Bad Bramstedt	1/1/1900						Y	N
G1 - Bad Meinersen	845	02400194510223	1/7/1915	CUSTOMER	Herr		Karl-Heinz	Herrberger	M	Wilhelmshöhe App. 122	23701	Eutin	1/1/1901						Y	N

Il report è basato su una tabella principale. La tabella può essere navigata sulla dimensione Sales Network, partendo dal livello più alto (Distretto) fino a livello più basso (Shop). In questo report non ci sono KPI. Successivamente verranno previste maggiori informazioni rispetto a quelle presenti in figura

- **Hearing Tests Analysis**

Questo report comprende le seguenti aree di analisi:

- Conta di Hearing Aid Test (navigabile per dimensione tempo);
- Conta dei clienti distinti con un Test dell'udito (si possono suddividere in clienti esistenti e nuovi clienti).

Il report si basa su una tabella che può essere navigata su tutte le dimensioni (ad esempio selezionando, filtrando o facendo drill-down) con le sole limitazioni sul livello massimo di profondità per ogni KPI. In particolare la tabella può essere navigata sulla dimensione temporale, con analisi trimestrali, mensili e settimanali, anche confrontando i risultati su anni diversi. KPI di questo report sono:

- I Test (numero di prove, il numero massimo di prove per cliente e il numero medio di test per cliente);
- I clienti con test (numero di clienti distinti che hanno fatto un test).

## 4.2 Analisi dello sviluppo progettuale

Nel seguente paragrafo descriveremo le attività progettuali svolte.

*Si è cercato di applicare il più possibile una logica di tipo Lean, fin dalle prime fasi del progetto, seguendo un approccio basato sul modello incrementale (vedi paragrafo 2.1.1).*

### 4.2.1 Stima delle attività

La stima delle attività e dei tempi concordati inizialmente ha subito un ritardo di 2 giorni), a causa di una variazione importante al progetto chiesta in fase di sviluppo e non stimabile inizialmente. Alla fine il progetto è stato chiuso nei tempi previsti con molta soddisfazione da parte del cliente.

La stima delle attività viene mostrata in figura seguente:

- **Analisi Soluzione.** È stata fatta l'analisi della soluzione proposta concordando le funzionalità che si volevano sviluppare ed eventuali difficoltà progettuali. Durante questa fase sono state prodotte una serie di demo dette POC(Proof of concept), utili a concordare aspetti grafici e stilistici del prodotto da sviluppare.
- **Modeling DataMart Analitico.** Ho costruito il modello analitico (DFM) con focus sui KPI e Dimensioni di Analisi.
- **Staging Area.** Ho importato le tabelle di interesse ed sviluppando successivamente la Staging area direttamente all'interno della nuvola (nel file Cloud) di QlikView.
- **Definizione Metadati.** Ho progettato e costruito la nuvola dei metadati del modello. In questa fase ho anche effettuato dei test di coerenza del modello e delle regole di aggregazione.
- **Sviluppo Layout Reporting.** Ho sviluppato il layout dei report, con l'aiuto del Client manager.
- **Realizzazione Reports e UAT.** Ho sviluppato esattamente i report richiesti nel documento di analisi funzionale. Inoltre ho eseguito dei test di coerenza sulle informazioni visualizzate, ma sono stati gli utenti finali ad approvare, durante il test finale (UAT), ed accettare il modello dati visualizzato. La gestione degli UAT

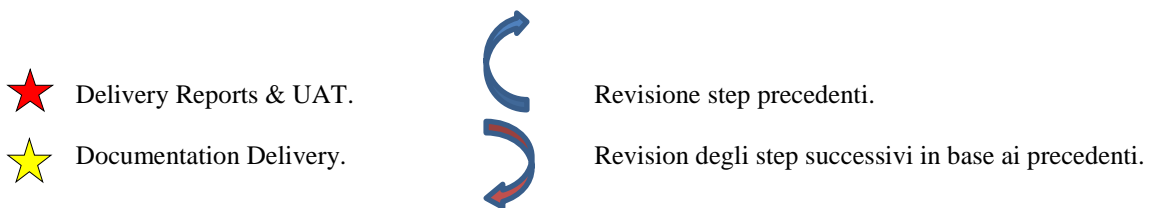
è stata a carico dell'IT Corporate. Gli UAT sono svolti in videoconferenza nella sede Amplifon Corporate di Milano, con gli utenti di test (un Utente Responsabile del Marketing e l'IT Specialist) di Amplifon De.

- **Documentazione Tecnica e Manuale Utente (Inglese).** Ho prodotto la documentazione tecnica relativa alla soluzione secondo le specifiche precedentemente concordate.
- **Formazione.** Sono stati introdotti agli utenti gli strumenti di analisi (QlikView) e sono stati illustrati i KPIs e i Report in un corso di 2gg fatto in sede Amplifon De ad Amburgo.

## 4.2.2 Il modello applicato (incrementale)

Di seguito viene rappresentato il modello incrementale applicato al progetto. Tra le varie fasi verranno rappresentati i cicli successivi ad ogni rilascio (UAT).

Funzionalità	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 11
<b>Solution Analysis</b>								
<b>Modeling DataMart</b>								
<b>Staging Area</b>								
<b>Metadata Definition</b>								
<b>Developing UAT Reports</b>								
<b>Technical Documentation</b>								
<b>Formation</b>								



Nella tabella sopra rappresentata si possono osservare esattamente i flussi delle attività svolte e come siano stati effettuati i 4 rilasci UAT<sup>5</sup>. La tecnica usata è basata sul rilascio incrementale di Realise del sw che andavano a risolvere i problemi (bug) riscontrati dalle versioni precedenti, includendo nuove funzionalità ad ogni rilascio.

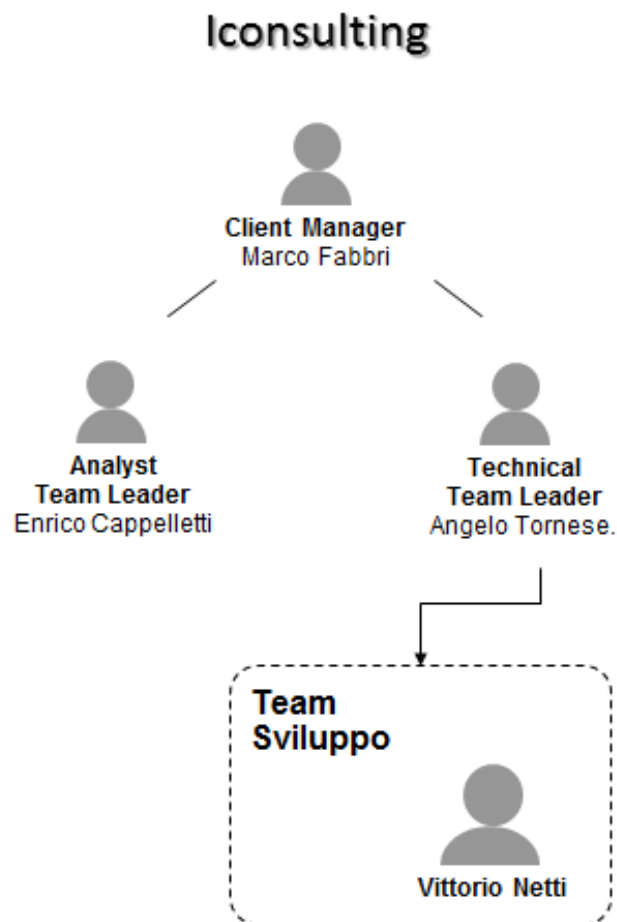
<sup>5</sup>(User Acceptance Test ) test di accettazione utente, una fase di verifica dello sviluppo software in cui l'utente finale valida la corrispondenza con i requisiti inizialmente espressi

### 4.2.3 I soggetti coinvolti

Al fine di avere una chiara visione dei soggetti coinvolti, dei ruoli e delle responsabilità di ognuno, si è concordato in fase iniziale una lista di figure, alle quali è stato attribuito un compito ed una responsabilità nel progetto.

#### *Iconsulting.*

Per quanto riguarda Iconsulting le figure coinvolte sono quelle rappresentate di seguito:



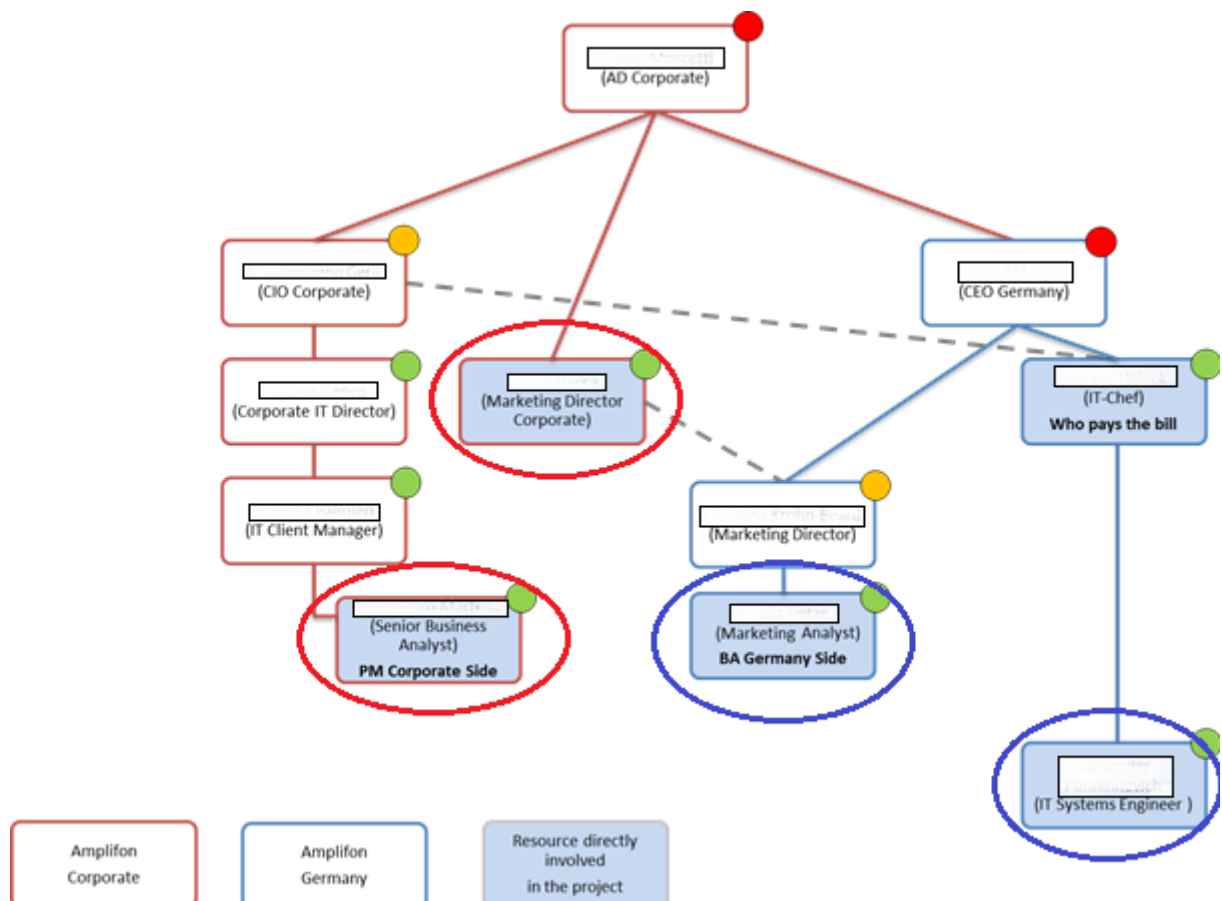
- Il *Client Manager*, che si è occupato della fase commerciale;
- L'*Analyst Team Leader*, che si è dedicato allo sviluppo dei rapporti con il cliente e al supporto nello sviluppo della componente grafica del sistema. Egli si è anche occupato della stesura della *documentazione funzionale*;
- Il *Technical Team Leader*, che ha dato il supporto nella fase commerciale (per le stime delle attività) e nella fase tecnica di sviluppo (supportando le prime fasi di analisi e costruzione del modello);



- *Il Team Sviluppo (Technical Developer), che si è dedicato all'attività pura di sviluppo e ha fornito il supporto tecnico all'Analyst nel momento delle richieste degli utenti. Il Team Sviluppo si è anche occupato della stesura della documentazione tecnica.*

### **Amplifon.**

Per quando riguarda Amplifon le figure coinvolte sono quelle rappresentate di seguito. Anche in questo caso, per motivi di Privacy, sono stati offuscati i nomi delle persone coinvolte, in quanto non abbiamo il diritto di pubblicare queste informazioni.



Le risorse coinvolte sono state inserite all'interno di mailing list dedicate, in modo da poter permettere continuamente una supervisione delle comunicazioni scambiate tra gli sviluppatori (di Iconconsulting) e gli utenti di test (di Amplifon De). Gli utenti di test

sono cerchiati di blu, mentre gli utenti supervisionanti sono cerchiati di rosso. Al termine delle attività, nella fase di formazione, si è deciso di responsabilizzare gli utenti di test dando loro l'opportunità di fare formazione ai propri colleghi, ovviamente supportati dal Team di sviluppo di Iconsulting. Oltre alla formazione pratica è stata fatta l'attività di formazione tecnica agli esperti IT, descrivendo la documentazione sviluppata e dando loro la possibilità di interpretarla, oltre a fornire le basi principali per l'auto manutenzione del sistema. In questo modo abbiamo creato la competenza tale da permettere l'autonomia del cliente nella gestione delle eventuali richieste che potrebbero sorgere successivamente.

### 4.3 Sviluppo del sistema

In questa fase descriviamo nel dettaglio le procedure adottate per lo sviluppo e l'implementazione del sistema precedentemente descritto.

Lo sviluppo del sistema è stato fatto sfruttando una macchina remota messa a disposizione da Amplifon, con i tutti software necessari per lo sviluppo.

Sw utilizzati:

- *QlikView Desktop v.10.0.* Questo è il sw di sviluppo dell'applicazione. La versione non è la più recente. Attualmente è la 11.2, ma si prevede in futuro di migrare a questa release. Per questo motivo abbiamo sviluppato il sistema facendo attenzione che la migrazione possa avvenire senza problemi (è stata testata su macchine con versione 11.2).
- *SQL Server Management Studio.* Esso è usato per connettersi al SIMDE ed effettuare le query pure sul DB. Inoltre il SQL Server Management Studio ci ha permesso di fare: sia il test del codice sql sviluppato per l'applicazione (vedi fase ETL01), sia il test di consistenza dei valori rappresentati nei Report.
- *Microsoft Excel 2003.* Esso è usato per analizzare ed esportare i dati elaborati in fase di analisi, in modo da avere riscontri su di essi con gli utenti di Test. Inoltre Microsoft Excel 2003 ci è servito per costruire i file di configurazione (vedi cartella Setting).

La pubblicazione dei file sul server QlikView è stata gestita dal reparto IT di Amplifon Corporate, rispettando quella che è la policy di Amplifon.

Di seguito andremo a descrivere per ogni fase la lista delle attività svolte.

<i>Fasi</i>	<i>Attività svolta</i>
<b>Solution Analysis</b>	Analisi Funzionale e Documentazione Funzionale
	Test sui dati
<b>Modeling DataMart</b>	Import Tabelle dal SIMDE
<b>Staging Area</b>	Staging Dimensione Customer
	Staging Dimensione Sales Network

	Staging Time & Campaign & Media Type
	Staging Contatti (Test, Trial, Sale, ... )
<b>Metadata Definition</b>	Costruzione Nuvola
<b>Developing UAT Reports</b>	Report 1 Hearing Tests Analysis
	Report 2 Campaign Reporting Area
	Report 3.1 Customer Hearing Aid
	Report 3.2 Media Type Reporting Distribution
	Report 4 Funnel and Conversion Rates
<b>Technical Documentation</b>	Documentazione Tecnica
<b>Formation</b>	Formazione Cliente

### 4.3.1 Solution Analysis

Durante questa prima fase l'attività svolta ha riguardato due aspetti principali:

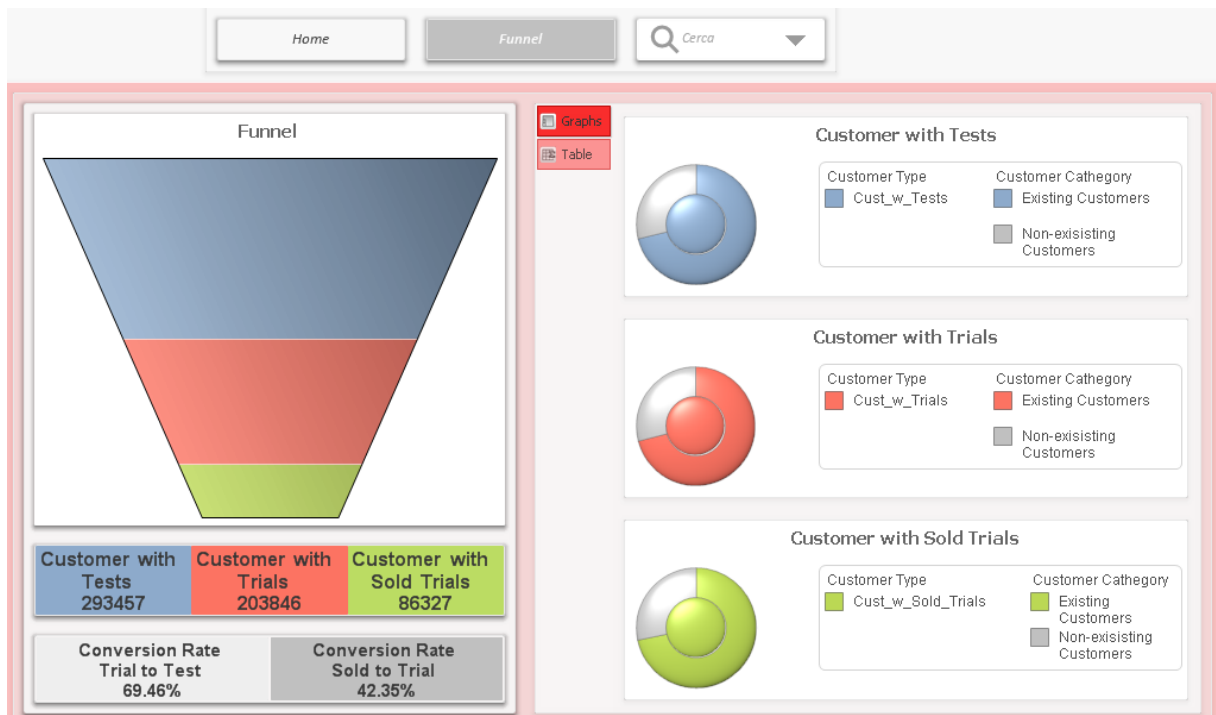
1. l'analisi funzionale;
2. i test sui dati.

#### *1. Analisi Funzionale*

Questo punto è stato portato avanti in stretta concomitanza col cliente, il quale rappresenta il maggior interessato agli aspetti funzionali del sistema. Per concordare le funzionalità sono stati sviluppati dei prototipi di modelli che sono serviti a dare l'idea di cosa e come il cliente voleva che l'applicazione visualizzasse le informazioni e, allo stesso tempo, ci hanno permesso di capire dove erano maggiormente concentrare le criticità del sistema, in modo da focalizzare maggiormente l'attenzione.

Per ogni report sono stati precedentemente costruiti dei prototipi utili a chiarire in questa fase gli aspetti precedentemente descritti. Ad esempio uno degli elementi critici di costruzione è stato il Funnel nella sua logica implementativa. La difficoltà maggiore è stata nel riuscire a costruire una logica ad innesco, in cui i clienti dovevano essere valutati in base al fatto di aver effettuato un test, nel periodo di test selezionato, e di

questi se avevano effettuato una trial, nel periodo trial selezionato, ed a loro volta, se avevano acquistato.



Come si può notare in questa immagine di test abbiamo ricostruito una possibile soluzione del report del funnel. Dobbiamo sottolineare che la costruzione dei prototipi è stata fatta in maniera rapida e agevole solo grazie all'estrema flessibilità del software di sviluppo QlikView. *(I dati sono completamente inventati)*

## 2. Test sui dati

In questa fase abbiamo effettuato dei test diretti sulla sorgente dati.

Tramite connessione remota al server di sviluppo di Amplifon, abbiamo potuto effettuare analisi dirette sui dati contenuti nel SIMDE tramite query in SQL.

È stata effettuata una prima fase di analisi di conformità dei dati e di ricerca delle tabelle da coinvolgere nel sistema.

Dall'analisi effettuata e da quello che ci è stato comunicato dai sistemisti, abbiamo preso atto che i dati presenti nel sistema non sono tutti pienamente consistenti. Una

migrazione del sistema fatta prima del 2010 ha cercato di porre rimedio a questo, ma a causa di:

- scarse segnalazioni da parte dei sistemisti interni,
- mancati inserimenti da parte dei negozi,
- perdita di informazioni storicamente precedenti al 2000 dopo la migrazione,

i dati minori al 2010 risultano non pienamente affidabili.

Questo è stato un limite del quale abbiamo dovuto tenere conto nelle analisi.

### 4.3.2 Modeling DataMart

Questa fase di sviluppo, durata circa 3-4 giorni, è stata impostata partendo dalle riflessioni e dai risultati dell'analisi emerse nella prima fase. Infatti in questa fase è stata impostata e sviluppata la parte di Connection and Extraction delle tabelle all'interno della fase di ETL01.

Le tabelle estratte sono le seguenti:

- CRM\_CAMPAIGN
- CRM\_PicklistValue
- CU0001
- CU0025
- CU0030
- CU0040
- CU0032
- DORG
- IC0002
- IC0001
- OP0002
- OP0003
- OP0402
- SY0016
- SY0024
- SY0025
- SY0044
- SY0049
- SY0223
- SY0224

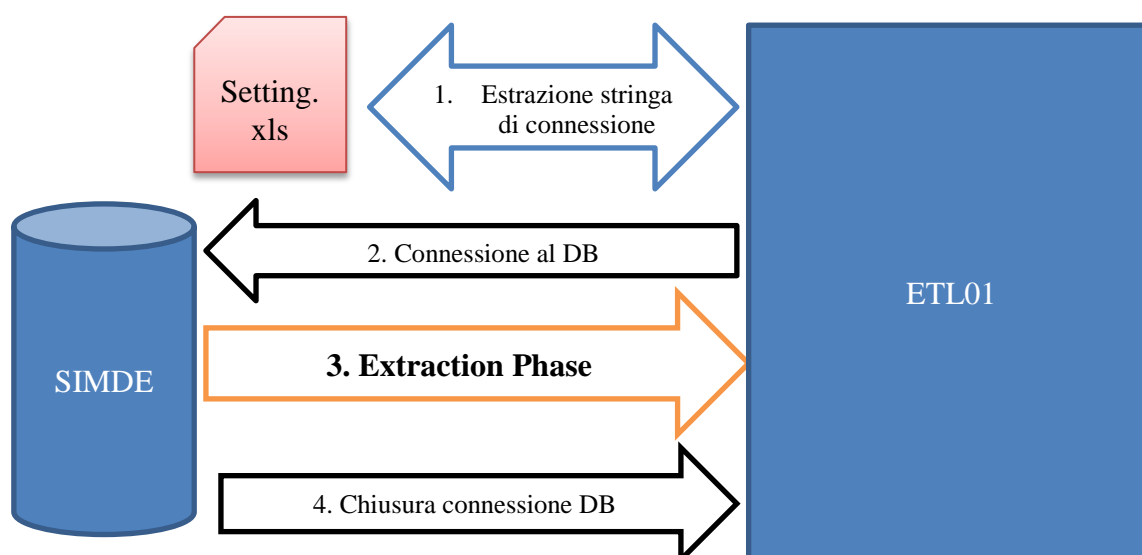
Per problemi di privacy non divulghiamo il significato delle singole tabelle, diciamo solo che tutte provengono dal SIMDE, comprese le prime due che, anche se provengono da un cubo multidimensionale, creato per un altro sistema di analisi per Amplifon DE, sono state importate nel SIMDE e aggiornate costantemente.

I dati estratti seguono logiche particolari, in base alle richieste fatte dal cliente.

Ad esempio le informazioni di vendita devono appartenere ad un insieme ben preciso di informazioni relativo a prodotti di tipo Hearing Aids.

Per effettuare l'estrazione dei dati in QlikView si effettua una connessione alla sorgente dati. La gestione tramite Best practices prevede che le operazioni di connessione vengano centralizzate all'interno di un file Excel e così è stato fatto. La stringa di connessione viene letta dal file Setting.xls e permette il collegamento al SIMDE, dal quale, tramite codice sql, viene fatta l'estrazione delle tabelle.

Nella figura seguente viene mostrato questo processo nelle sue varie fasi:



Come si nota chiaramente l'estrazione avviene una sola volta nella fase di ETL01, questo significa che in questo modo le operazioni di estrazione vengono centralizzate in unico file secondo una specifica sequenza (vedi Staging Area), in modo tale da facilitare eventuali gestioni e manutenzioni del sistema. Il tecnico manutentore saprà esattamente che in caso di problemi di estrazione dovrà controllare esclusivamente quel file senza preoccuparsi degli altri (QlikView best practices).



### 4.3.3 Staging Area

Durante la fase di staging verranno effettuate le operazioni di costruzione del modello interessato, secondo le dimensioni e le gerarchie che abbiamo descritto in precedenza. Questa è la fase con maggior criticità per l'intero progetto, in quanto viene interessata non solo per le prime operazioni di costruzione del modello, ma anche per tutte le altre operazioni di modifica successive. Infatti, dopo il rilascio di ogni report, le modifiche e obiezioni del cliente ci hanno portato ad effettuare modifiche su questo livello (logica ciclica). Per questo motivo si è deciso di investire maggior tempo per organizzare al meglio e nel modo più chiaro possibile il codice di staging.

La lista delle operazioni di staging è la seguente:

- Staging Customer
- Staging Sales Network
- Staging Time
- Staging Campaign
- Staging Media Type
- Staging Contatti (Contact Type, Historical Contact e Contact )

Le operazioni di Staging sono state effettuate nelle due fasi di ETL:

- ETL01 fase principale di staging in concomitanza con le operazioni di estrazione dei dati dal DB.
- ELT02 fase secondaria di staging, che prevede principalmente l'operazione di rinomina delle tabelle e dei campi.

#### ***ETL0:1 Prima fase di Staging***

Questa fase prevede la parte più interessante della fase di Trasformazione dei dati, in cui le operazioni di modifica sono state effettuate sfruttando codice SQL, molto più flessibile per determinate operazioni (es. group by, query annidate, ecc.) rispetto al codice Qlikview. Per questo motivo il file ETL01 include al suo interno porzioni di

codice all'interno di blocchi SQL. Nella figura sottostante si mostra il codice di esempio di estrazione e trasformazione della dimensione Media Type:

```

//*****
//*****
//** FOLDER USED FOR DEFINE S_MEDIA
//*****
//*****

SUB S MEDIA;
S_MEDIA:
SQL
----Media Dimension-----

        SELECT
        ISNULL(CODSOC,'024') AS CODSOC,                --Company code
        ISNULL(CODTICO,'-1') AS CODTICO,              --Media code
        MAX(ISNULL( UPPER(DSTICO), 'Not defined')) AS DSTICO --Media description

        FROM dbo.SY0024 SY24
        GROUP BY
                ISNULL(CODSOC,'024'),
                ISNULL(CODTICO,'-1')

        --- Add a dummy row for identify the undefined media type
        UNION

        SELECT '024' AS CODSOC,
                '-1' AS CODTICO,
                'Not defined' AS DSTICO

;

STORE S_MEDIA INTO [$(SubPathQVD)S_MEDIA.qvd] (qvd);
DROP Table S_MEDIA;
ENDSUB;

```

Come si nota dal codice, sono state effettuate operazioni di normalizzazione dei dati, gestione dei campi mancanti, gestione delle anomalie, ecc.

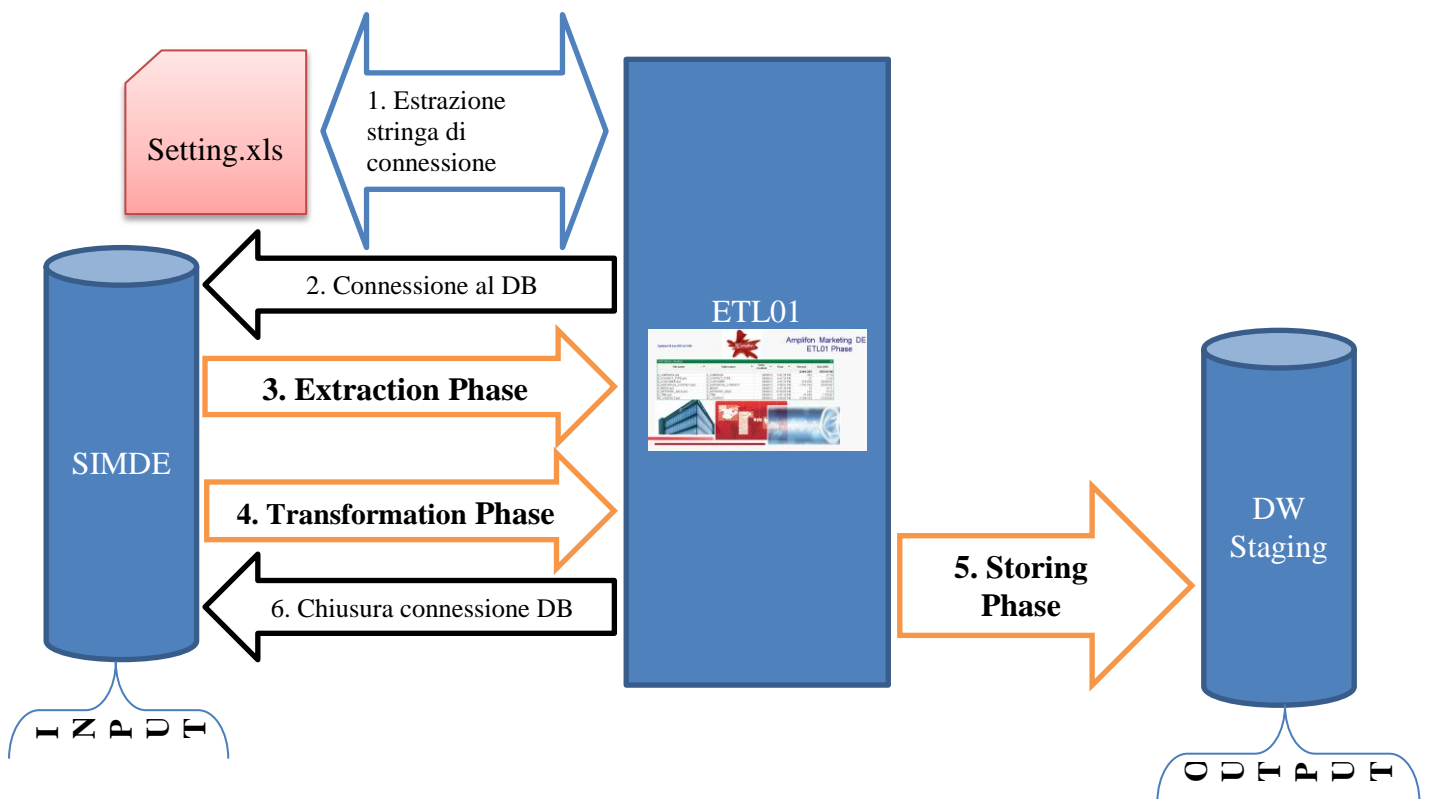
Questa è la più semplice delle operazioni sulle dimensioni, in quanto prevede semplicemente la normalizzazione e la gestione dei campi mancanti o nulli da una tabella presente nel SIMDE. Operazioni molto più complesse sono state effettuate per costruire la tabella dei fatti F\_Contact, la quale prevede una serie di step preliminari per individuare gli elementi fondamentali del business.

Seguendo la logica di naming aziendale, la fase di ETL01 produce delle tabelle di staging che hanno i seguenti suffissi:

- S\_: per indicare una dimensione (es. S\_MEDIA)
- SF\_: per indicare un fatto (es. SF\_CONTACT).

Il flusso delle operazioni, compute in fase di ETL01, è il seguente:

1. Lettura della stringa di connessione dal file di setting (di configurazione);
2. Connessione al SIMDE;
3. Fase di Estrazione dei dati (Extraction Phase);
4. Fase di Trasformazione dei dati (Transformation Phase);
5. Fase di salvataggio delle tabelle nella cartella DW\_Staging (Storing phase);
6. Chiusura connessione.



Nel file ETL01 viene anche mostrata la situazione delle tabelle generate all'interno della cartella DW\_Staging. Ovviamente questo ha il solo scopo di informare i tecnici sulle dimensioni e le tabelle coinvolte ma a in questo caso ci è utile per capire quali sono le tabelle generate da questa fase:

Updated 28 Jun 2013 at 12:06



## Amplifon Marketing DE ETL01 Phase

File name	Table name	Data creation	Time	Record	Size (KB)
S_CAMPAIN.qvd	S_CAMPAIN	26/06/13	4:47:25 PM	8.461.322	125.917,98
S_CONTACT_TYPE.qvd	S_CONTACT_TYPE	26/06/13	4:47:32 PM	293	27,19
S_CUSTOMER.qvd	S_CUSTOMER	26/06/13	4:47:22 PM	27	12,43
S_HISTORICAL_CONTACT.qvd	S_HISTORICAL_CONTACT	26/06/13	4:47:22 PM	518.976	30.600,57
S_MEDIA.qvd	S_MEDIA	26/06/13	4:48:51 PM	1.791.019	20.874,97
S_NETWORK_SALE.qvd	S_NETWORK_SALE	26/06/13	4:47:26 PM	19	10,11
S_TIME.qvd	S_TIME	28/06/13	12:59:03 PM	220	15,23
SF_CONTACT.qvd	SF_CONTACT	26/06/13	4:47:32 PM	41.450	1.760,67
		26/06/13	4:55:02 PM	6.109.318	72.616,83



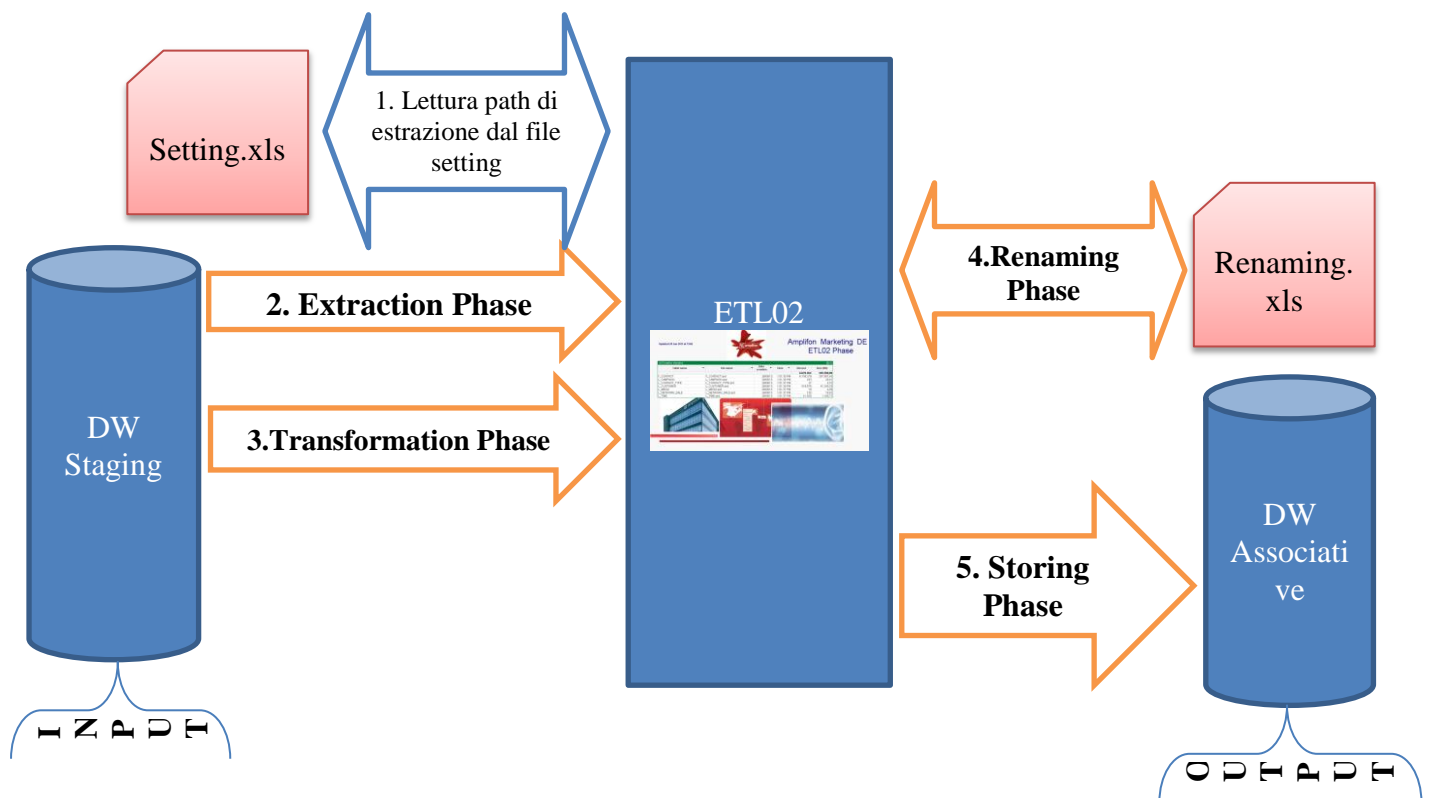
### *ETL02: Seconda Fase di Staging*

In questa seconda fase le operazioni che vengono effettuate riguardano due casi:

- Elaborazioni base delle tabelle: filtraggi particolari (casi richiesti dal cliente) e azioni di fusione delle tabelle dei fatti;
- Rinomina:
  - Rinomina dei nomi delle tabelle (centralizzata dal file Renaming)
  - Rinomina dei campi delle tabelle (centralizzata dal file Renaming)

Il flusso delle operazioni, compute in fase di ETL02, è il seguente:

1. Lettura del path di estrazione dal file di setting;
2. Estrazione delle tabelle da DW\_Staging (Extraction Phase);
3. Fase di Trasformazione dei dati (Transformation Phase);
4. Fase di Rinomina dei dati (Renaming Phase);
5. Fase di salvataggio delle tabelle nella cartella DW\_Associative (Storing phase);



Seguendo la logica di naming aziendale, la fase di ETL02 produce delle tabelle di staging che hanno i seguenti suffissi:

- L\_: per indicare una dimensione (es. L\_MEDIA)
- F\_: per indicare un fatto (es. F\_CONTACT).

Per permettere il funzionamento della logica di QlikView, i nomi delle chiavi delle tabelle sono stati rinominati allo stesso modo, così nella fase di costruzione della nuvola le tabelle potranno collegarsi con fra loro automaticamente.

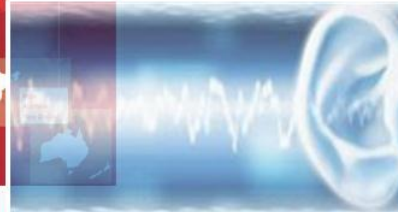
Come per il file ETL01 anche in ETL02 viene mostrata la situazione delle tabelle generate all'interno della cartella DW\_Associative. Anche in questo caso, di seguito abbiamo una rappresentazione delle suddette tabelle:

Updated 28 Jun 2013 at 13:06



## Amplifon Marketing DE ETL02 Phase

QVD tables situation						
Table name	File name	Data creation	Time	Record	Size (KB)	
F_CONTACT	F_CONTACT.qvd	28/06/13	1:01:35 PM	6.109.379	297.967,45	
L_CAMPAIN	L_CAMPAIN.qvd	28/06/13	1:01:38 PM	293	26,61	
L_CONTACT_TYPE	L_CONTACT_TYPE.qvd	28/06/13	1:01:37 PM	27	4,70	
L_CUSTOMER	L_CUSTOMER.qvd	28/06/13	1:01:38 PM	518.976	41.240,29	
L_MEDIA	L_MEDIA.qvd	28/06/13	1:01:37 PM	19	4,69	
L_NETWORK_SALE	L_NETWORK_SALE.qvd	28/06/13	1:01:37 PM	220	18,62	
L_TIME	L_TIME.qvd	28/06/13	1:01:37 PM	41.450	1.105,73	



#### 4.3.4 Metadata Definition

In questa fase verrà effettuata la costruzione della Nuvola di QlikView, in base alle logiche e alle restrizioni imposte dal Sw.

La nuvola di QlikView non è altro che il modello di metadati generato dal software e che gli permette di effettuare le elaborazioni sui dati. La nuvola viene interamente caricata in memoria principale così da permettere rapidità ed efficienza nell'elaborazione dei dati.

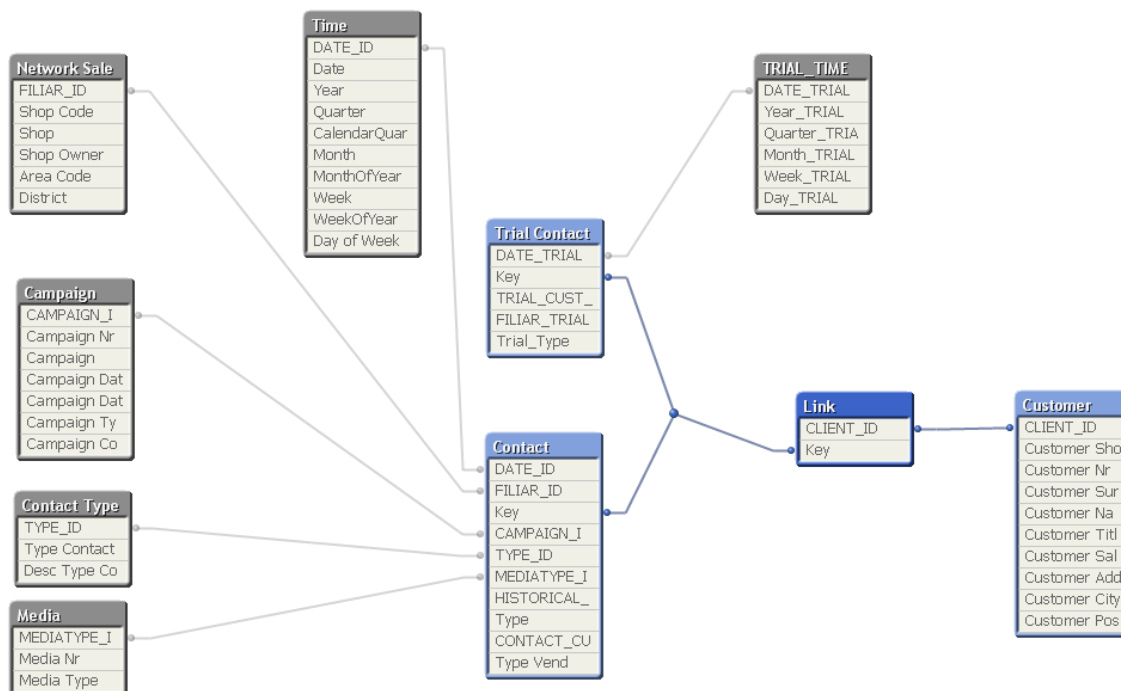
Il file *Cloud* è quello interessato dalle operazioni di questa fase. Infatti in questo file vengono gestite tutte le logiche necessarie alla creazione della nuvola.

Seguendo le Best practices del produttore del software è stata costruita una tabella di unione, detta Link Table, che permetto di risolvere i conflitti generati da cicli. Infatti per rispondere alla logica del funnel si è deciso di generare delle tabelle dei fatti fittizie per le Trial, che condividono la dimensione Customer.

Il ciclo generato dalla tabella fittizia è stato gestito, appunto, tramite una Link Table da noi generata, evitando che il Sw risolvesse il conflitto con delle *chiavi sintetiche* che risultano poco affidabili e sono sconsigliate dal produttore.

Come si nota dall'immagine sottostante, la parte della nuvola colorata rappresenta:

- a sinistra i fatti;
- a destra la dimensione condivisa.



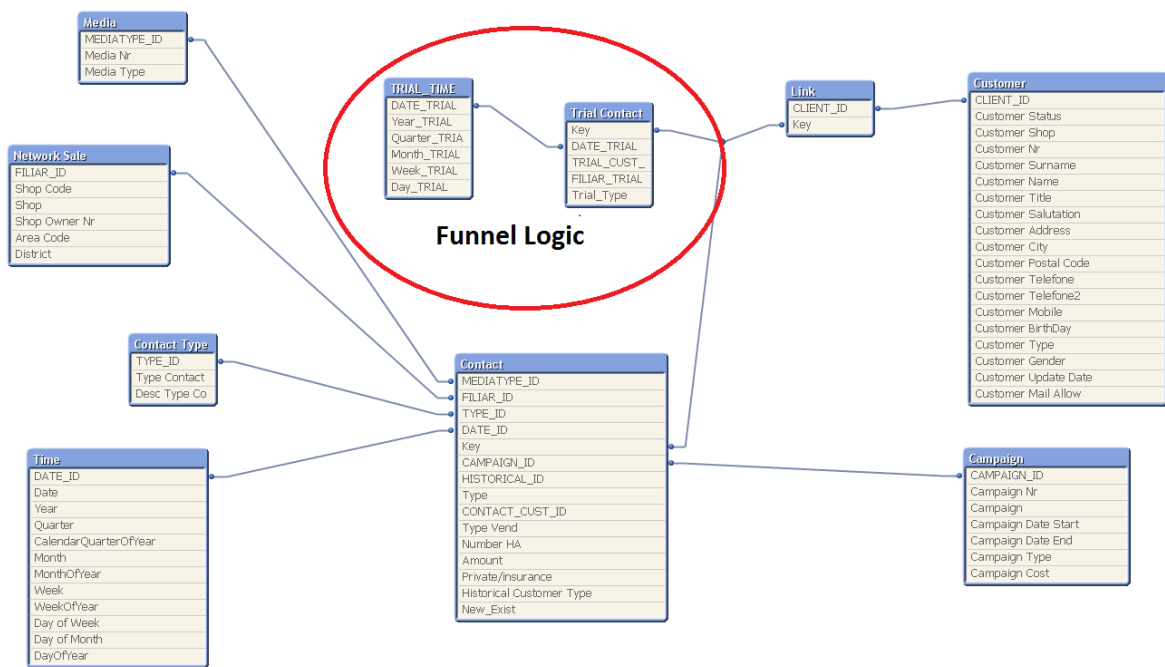
Oltre a questo è stato previsto, in un tab specifico del file Cloud.qvw, uno spazio disponibile per la generazione di tabelle manuali (*Inline Table* in gergo QlikView). Questo risulta utile per il reparto IT che prenderà in gestione il sistema e che potrà con pochi sforzi aggiungere o eliminare le nuove tabelle richieste (operazioni possibili dopo aver effettuato il corso di *formazione tecnico*).

Per terminare la procedura di *Metadata Definition*, è stata effettuata l'ultima procedura di rinomina delle tabelle, in modo da assegnare nomi significativi per ognuna, privi dei suffissi.

Alla fine la nuvola generata con tutte le dimensioni equivale ad uno Star schema in cui le dimensioni sono portate al livello gerarchico più basso e i fatti sono aggregati in unico fatto centrale, ad esclusione della logica per il funnel (Trial Cotact+Trial Time).

La nuvola è la seguente:

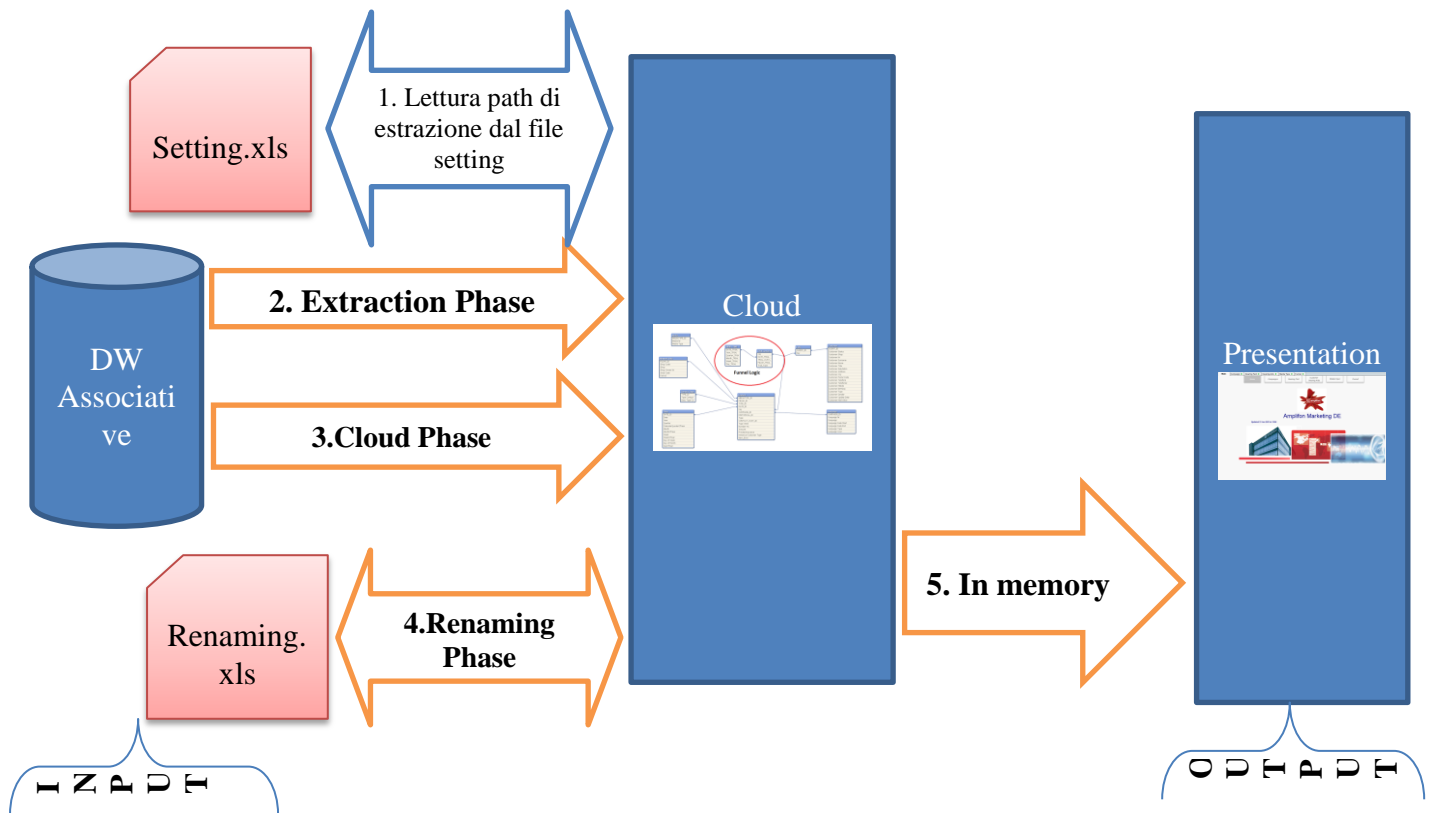




Dimensioni e fatti rispecchiano esattamente le gerarchie definite al paragrafo 4.1.3, ad esclusione della dimensione e del fatto per la logica del funnel.

Il flusso delle operazioni, compute in fase di Cloud, è il seguente:

1. Lettura path di estrazione dal file setting;
2. Estrazione delle tabelle da DW\_Associative (Extraction Phase);
3. Fase di costruzione della nuvola (Cloud Phase);
4. Fase di Rinomina delle tabelle (Renaming Phase);
5. Conservazione dei dati in memoria principale (In memory);



Il file cloud non salva i dati in cartelle specifiche ma li conserva in memoria perché verranno usati dal presentation file (quello che conterrà i Report).

### 4.3.5 Developing UAT Reports

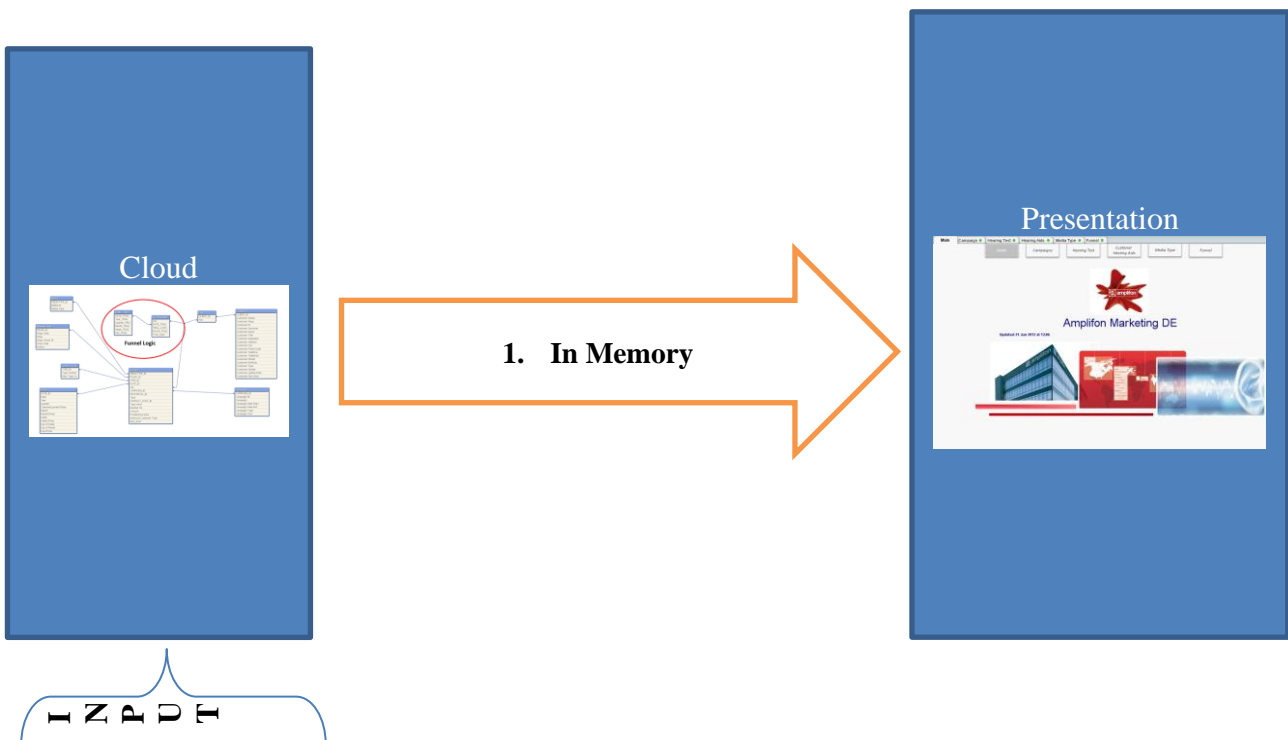
Lo sviluppo dei report è stato il momento in cui l'utente ha potuto tangere con mano i risultati del lavoro, ed insieme ad esso è riuscito ad addentrarsi meglio nella logica del sistema IT che non è la stessa al quale era abituato a pensare.

Questa fase ha generato il massimo livello di impegno sul progetto, in quanto, con rilasci settimanali, si è provveduto a rilasciare report nuovi e allo stesso tempo a recuperare e risolvere bug o problemi relativi ai report precedenti (scoperti dagli utenti di test).

Lo sviluppo dei report avviene all'interno di un file che contiene solo l'aspetto di presentazione (prova di codice) e che include il file Cloud.qvw che mantiene la logica del sistema. In questo modo si potrà garantire il flusso parallelo di eventuali nuovi progetti, che sfruttano la stessa logica implementativa, ma rappresentano informazioni diverse.

Il flusso delle operazioni, compute in fase di Cloud, è il seguente:

1. Inclusione della logica e dei dati (In memory)



I report sviluppati sono 5 e sono stati rilasciati in 4 UAT:

- Report 1: Hearing Tests Analysis
- Report 2: Campaign Reporting Area
- Report 3: Customer Hearing Aid
- Report 4: Media Type Reporting Distribution
- Report 5: Funnel and Conversion Rates

Struttura dei report

Ad esclusione della Home page, tutti i report si compongono di una struttura di filtri fissa (ad eccezione del report funnel che include un filtro sulla dimensione Date Trial).

The screenshot displays the Amplifon reporting dashboard. At the top, a navigation bar includes tabs for Home, Campaigns, Hearing Tests (selected), Customer Hearing Aids, Media Type, and Funnel, along with a search icon. Below the navigation bar, there are two calendar widgets: one for '20/04/2000' with a 'Clear Calendar' button, and another for '17/06/2013' with an 'Apply Calendar' button. The left sidebar contains several filter panels: 'Time' (Year, Quarter, Month, Week), 'Sales Network' (District, Shop), 'New / Existing' (Existing Customer, New Customer), 'Campaign Type' (Advertising - Bulk, Advertising - Call C..., Advertising - Maga..., Advertising - Newsp..., Mailing - Direct Mar..., Mailing - Emailing, Mailing - Promotion..., National Campaign, Not Defined), 'Contact Type' (Appointment Check..., Fitting, New Appointment, Other, Quick Sale, Registration, Sale, Test, Trial), and 'Media Type' (AMPLIBUS, BULK MAIL, CALL CENTRE, CONSUMER REFER..., DIRECT MAIL, ENT / PHYSICIAN ..., INTERNET, MAGAZINE, NEWSPAPER). The central area is a large white box with the text 'Riquadro relativo al report'. At the bottom, a 'Current Selection' panel shows 'Year' set to '2013'.

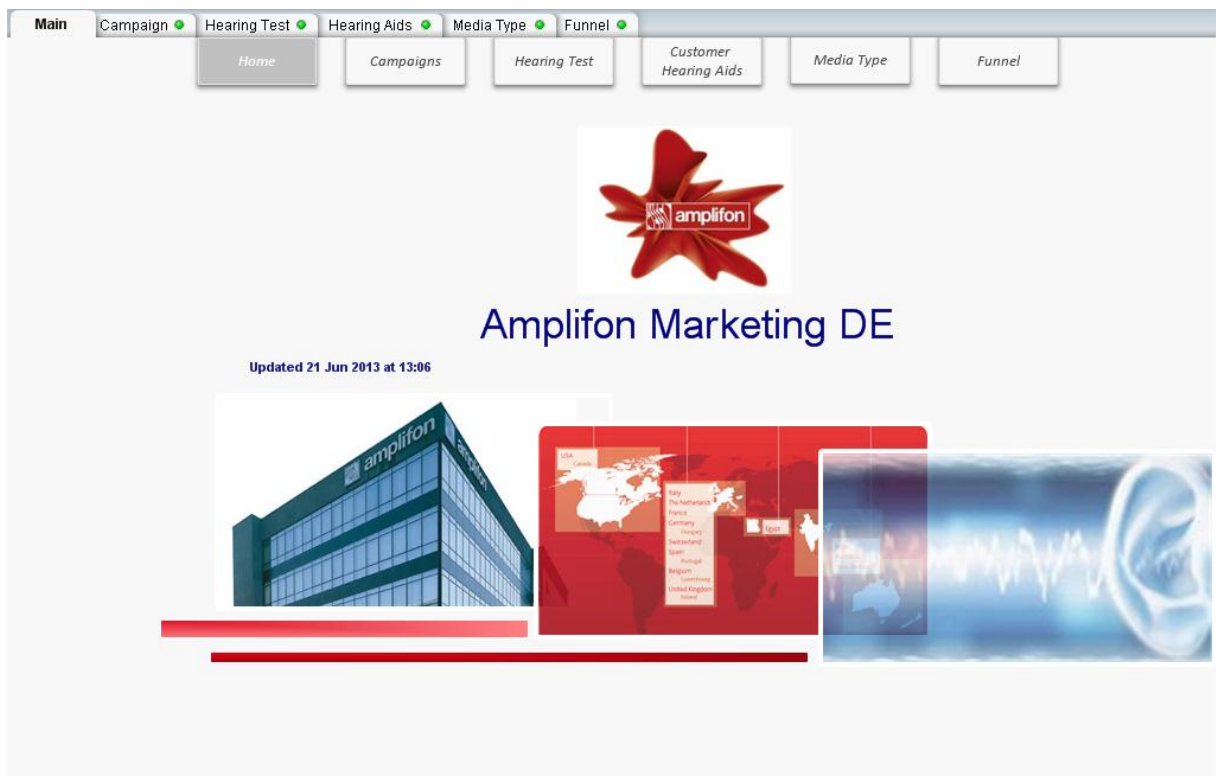
Come si nota, escludendo il riquadro del report, posto in forma centrale, sono presenti tre zone ben distinte:

- Filtri: i filtri riguardano principalmente tutte le dimensioni del modello, ed hanno la possibilità di essere applicati sul singolo elemento o su di una selezione multipla. Il filtro temporale può essere applicato o dal calendario, oppure selezionando direttamente anni, quartili, mesi e settimane.

- Current Selection: in basso la current selection ci dà l'informazione relativa ai filtri applicati
- Report navigation & Search Button: in alto l'utente può navigare fra i report o può effettuare una ricerca globale su qualsiasi informazione.

## Home Page

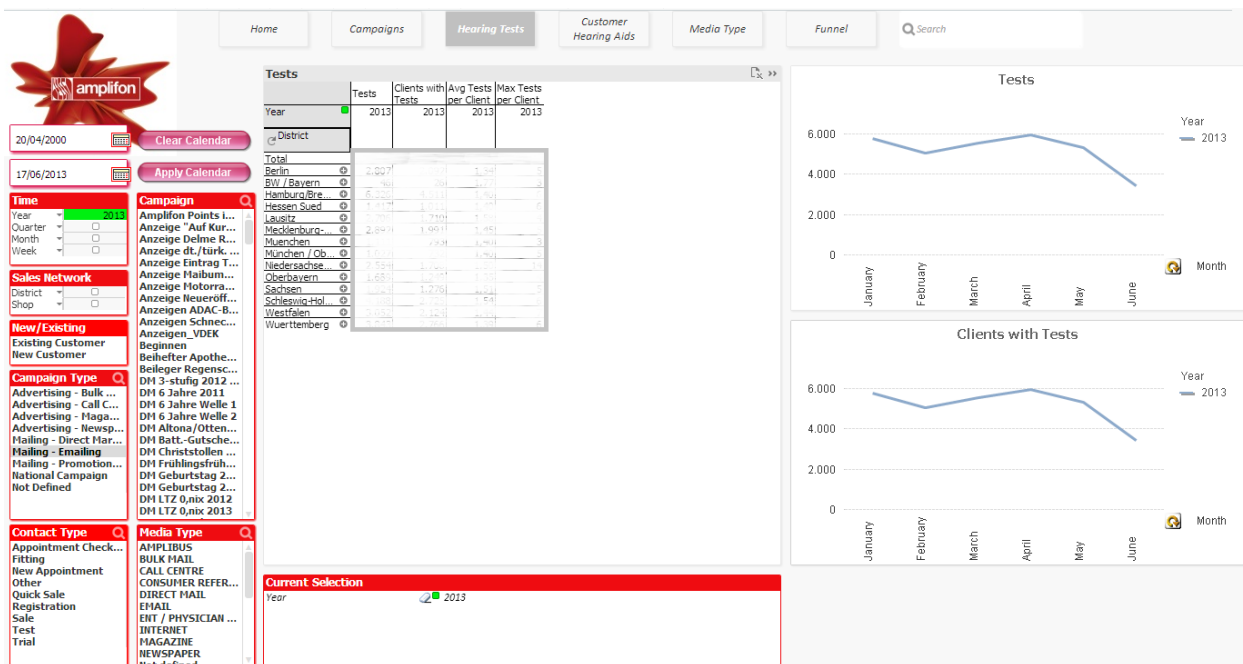
La Home page si compone di 6 bottoni di navigazione (1 per la home e 5 dei singoli report)



In questa pagina l'utente non può fare altro che entrare direttamente nel report interessato e navigarlo. La sola informazione utile riguarda la data di aggiornamento dei dati nel sistema.

## Hearing Tests

Primo dei report sviluppati, in cui l'utente ha la possibilità di fruire le informazioni statiche relative ai KPI di test.

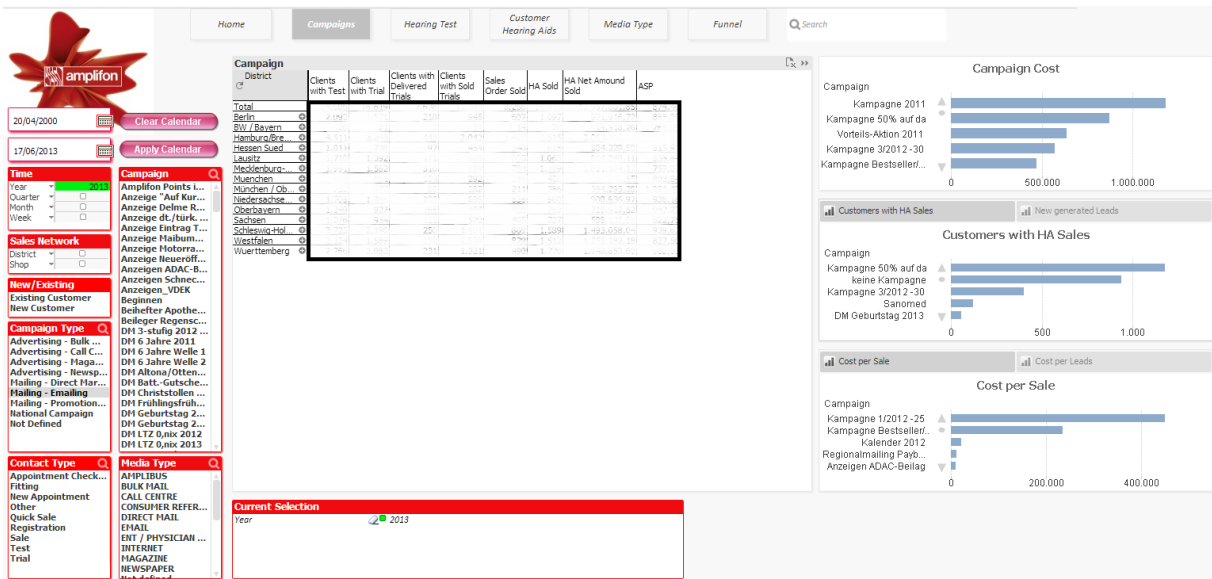


KPI visualizzati in questo report sono descritti al paragrafo 4.1.5 nella sezione relativa al report Hearing Test.

Grafici a linea: distribuzione dei test e dei clienti con test per intervalli temporali variabili (settimanale, mensile, quadrimestrale).

### Campaign Reporting Area

Il report Campaign permette di visualizzare informazioni di distribuzione delle campagne, in base alla distribuzione territoriale (Network sales). Le informazioni visualizzate rispettano la logica del CRM e non del FUNNEL, perciò i test, trial e sales non sono legati fra loro (dalla dimensione Cliente).



KPI visualizzati in questo report sono descritti al paragrafo 4.1.5 nella sezione relativa al report Campaign.

Osservazioni: il seguente report ha avuto criticità nel momento in cui si doveva definire la logica di calcolo dei Customer Hearing Aids. Infatti per estrarre questa informazione abbiamo operato diverse modifiche nella fase di ETL01. La difficoltà principale risiedeva nel fatto che per estrarre l'informazione relativa al corretto dispositivo acustico venduto si doveva tenere conto di Sales filtrate per:

- No CR (Credit Note che annullano una vendita)
- Tipo dispositivo= Hearing Aid (campi particolari di tabelle coinvolte)
- Codice merce="xxx" (codice merce che individua gli HA)
- Quantità vendute>0 (quantità diverse rappresentano altre info)
- Sottrazione di sconti dal totale.

Queste informazioni erano inizialmente poco chiare al reparto IT, e sono state individuate nel corso del progetto, tramite la nostra collaborazione.

Questo ha comportato, come solitamente avviene per questi progetti, revisioni continue al modello, anche dopo il rilascio di report successivi. Alla fine l'informazione desiderata è stata raggiunta ed è stata correttamente tracciata e approvata da tutti i reparti.

**Grafici a barre:** distribuzione dei valori di Costo campagna, numero clienti che hanno acquistato (nuovi e vecchi), costo campagna per clienti (nuovi e vecchi). Questi valori sono espressi per singola campagna promozionale.

## Customer Hearing Aid

In questo report vengono visualizzate le informazioni di dettaglio dei singoli clienti. Il report ha lo scopo di dare informazioni anagrafiche del cliente e su quale negozio il cliente ha acquistato. Per privacy sono state mascherate le info legate strettamente al cliente

Shop	Customer	Customer No	Customer Type	Title	Salutation	Name	Surname	Gen	Street	Postal Code	City	First Last	Second Last	Third Last	Hall Allowed	No Call Flag	Author. Provided
Rottach-Eoern	17/02/1932	40801075	CL	Dr.	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	14/02/2013	13/09/2012	14/08/2012	N	Y	
Chemnitz	19/04/1930	12900291	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	13/03/2013	06/11/2012	08/06/2012	N	Y	
Hamburgo-Berendorf	28/09/1949	93112754	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	09/04/2013	02/04/2013	16/05/2012	N	Y	
Wolfenbüttel	16/04/1936	957000CF	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	17/05/2013	15/05/2012	03/02/2012	N	Y	
Nördlingen	31/03/1928	60112664	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	18/09/2012	31/07/2012	07/12/2011	N	Y	
Hamburgo-City	14/02/1937	91232557	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	10/09/2012	02/12/2011	01/11/2011	N	Y	
Aalen	19/07/1954	6023017C	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	18/10/2012	13/07/2012	26/10/2011	N	Y	
Dachau	03/08/1915	40200589	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	14/12/2011	28/11/2011	20/10/2011	N	Y	
Berlin-Friedrichsfeld	06/10/1929	1310113K	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	06/07/2012	12/04/2012	29/06/2011	N	Y	
Cottbus-Furst-Puck	21/10/2009	1050023A	CL	ND	An die Eltern	[Redacted]	[Redacted]		[Redacted]	[Redacted]	[Redacted]	08/12/2012	23/04/2012	29/06/2011	N	Y	
Hamburgo-Rahlstedt	28/03/1936	9190126C	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	09/12/2011	15/07/2011	28/06/2011	N	Y	
Gießen	29/05/1930	32102361	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	22/04/2013	02/11/2012	16/05/2011	N	Y	
Berlin-Zehlendorf	04/02/1940	92880846	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	18/03/2013	14/12/2011	14/03/2011	N	Y	
Berlin-Tempelhof	09/01/2005	91862631	CL	ND	An die Eltern	[Redacted]	[Redacted]		[Redacted]	[Redacted]	[Redacted]	05/12/2012	24/08/2011	12/01/2011	N	Y	
Anzeige Heuserriff...	29/01/1940	90890013	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	07/02/2013	19/11/2012	21/12/2010	N	Y	
Anzeige Motorra...	14/04/1937	91446203	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	07/02/2013	19/11/2012	21/12/2010	N	Y	
Anzeige Neuerriff...	31/01/1942	10201064	CL	ND	Herr	[Redacted]	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	11/04/2013	10/04/2012	04/11/2010	N	Y	
Anzeigen ADAC-B...	19/03/1930	96000117	CL	ND	Frau	[Redacted]	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	29/05/2012	16/11/2011	22/10/2010	Y	N	
Anzeigen Schme...	Berlin-Kreuzberg	20/06/1988	13301077	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	03/01/2013	09/11/2010	29/09/2010	N	Y	
Anzeigen JVDEK	Berlin-Kreuzberg	28/04/1918	14600046	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	13/12/2010	01/11/2010	02/09/2010	N	Y	
Beginnen	Hannover-Großfeld	20/11/1945	32102361	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	09/03/2012	03/03/2011	23/07/2010	N	Y	
Beihelfer Apothe...	Berlin-Kreuzberg	15/07/1923	98301772	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	19/09/2012	03/08/2012	14/07/2010	N	Y	
Belteger Regens...	Malchin	12/08/1957	94100747	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	27/08/2012	15/11/2010	19/05/2010	N	Y	
DH 5-Stufig 2012 ...	Osnabrück-City	02/08/1928	97960925	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	23/02/2013	30/04/2012	28/04/2010	N	Y	
DH 6 Jahre 2011	Berlin-Tempelhof	09/02/1931	91886448	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	14/05/2013	30/05/2012	07/04/2010	N	Y	
DH 6 Jahre Welle 1	Senftenberg	28/12/1936	10901209	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	13/08/2012	23/06/2010	11/01/2010	N	Y	
DH 6 Jahre Welle 2	Hamburgo-Mörfeld	03/05/1920	92004866	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	11/07/2012	12/07/2011	03/12/2009	N	Y	
DH Altona/Otten...	Bad Schwarzbau	15/12/1923	92232141	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	23/05/2012	02/05/2012	24/11/2009	N	Y	
DH Battl-Gutsche...	Berlin-Lichtenrade	02/05/1934	93442596	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	05/04/2011	07/03/2011	10/11/2009	N	Y	
DH Christstollen ...	Piene	25/10/1936	96400580	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	07/03/2013	07/03/2011	10/11/2009	N	Y	
DH Frühlingsfrüh...	Hoyerswerda-Neus...	13/01/1939	10601801	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	04/10/2010	27/10/2009	26/10/2009	N	Y	
DH Geburtstag 2...	Neumünster	17/08/1927	90100234	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	06/02/2013	06/01/2011	21/10/2009	N	Y	
DH LTZ 0,nix 2012	Bad Eisen	05/02/1935	97816468	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	14/06/2012	04/06/2012	15/10/2009	N	Y	
DH LTZ 0,nix 2013	Elsterwerda	30/03/1928	11101497	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	13/05/2013	07/11/2012	30/09/2009	N	Y	
National Campaign	Berlin-Kreuzberg	27/08/1929	13301064	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	04/09/2012	06/08/2012	28/09/2009	N	Y	
Not Defined	Hamburgo-Krochdorf	17/08/1997	91223780	CL	ND	An die Eltern	[Redacted]		[Redacted]	[Redacted]	[Redacted]	14/06/2012	04/06/2012	15/10/2009	N	Y	
	Hamburgo-Nendorf	24/12/1926	92004532	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	22/02/2013	15/03/2012	24/08/2009	N	Y	
	Demmin	17/01/1960	93801119	CL	ND	Frau	[Redacted]	F	[Redacted]	[Redacted]	[Redacted]	15/06/2012	19/07/2011	21/08/2009	Y	N	
	Stutrow-Mitte	06/05/1960	15901118	CL	ND	Herr	[Redacted]	M	[Redacted]	[Redacted]	[Redacted]	16/05/2013	07/02/2013	21/08/2009	N	Y	

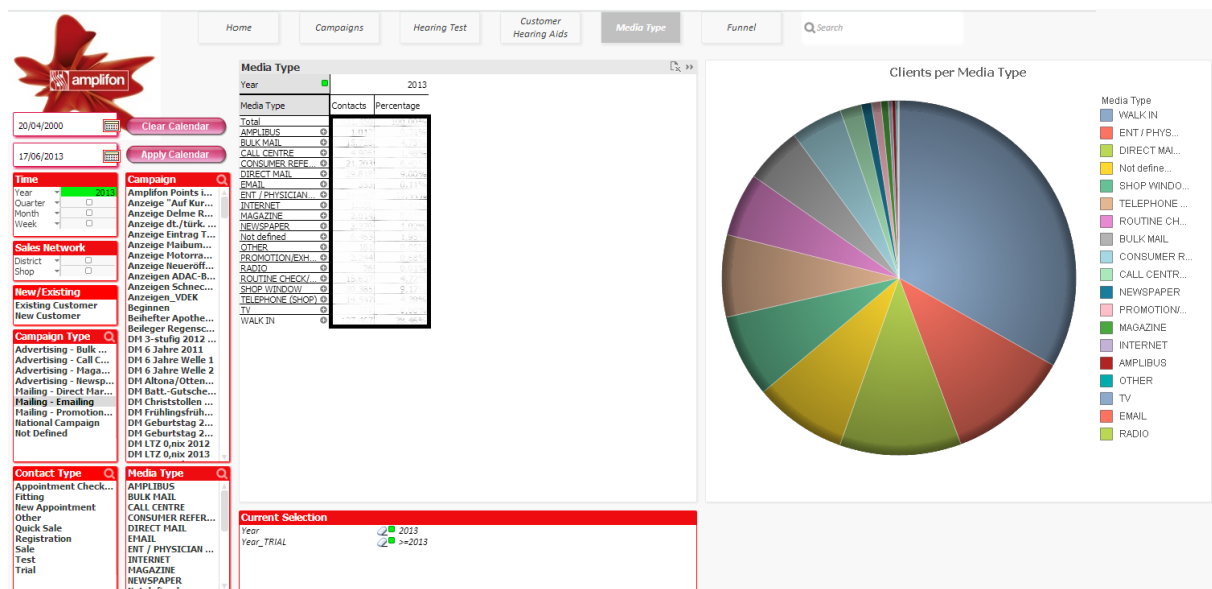
KPI non sono presenti in questo report ma per maggiori informazioni si rimanda al paragrafo 4.1.5 nella sezione relativa al report Customer Hearing Aid.

Le sole informazioni critiche rappresentano le date relative agli ultimi tre acquisti del cliente. Il calcolo è stato fatto dal lato Presentation, utilizzando la Set Analysis riducendo così le operazioni di Staging (caso in cui i campi vengano calcolati in fase ETL01).



## Media Type

Nel seguente report vengono visualizzate le informazioni relative ai Media Type, calcolando in numero di contatti (totali o suddivisi per tipo: es. Test, Trial, Sales, ecc.) avuti per ogni Media Type.

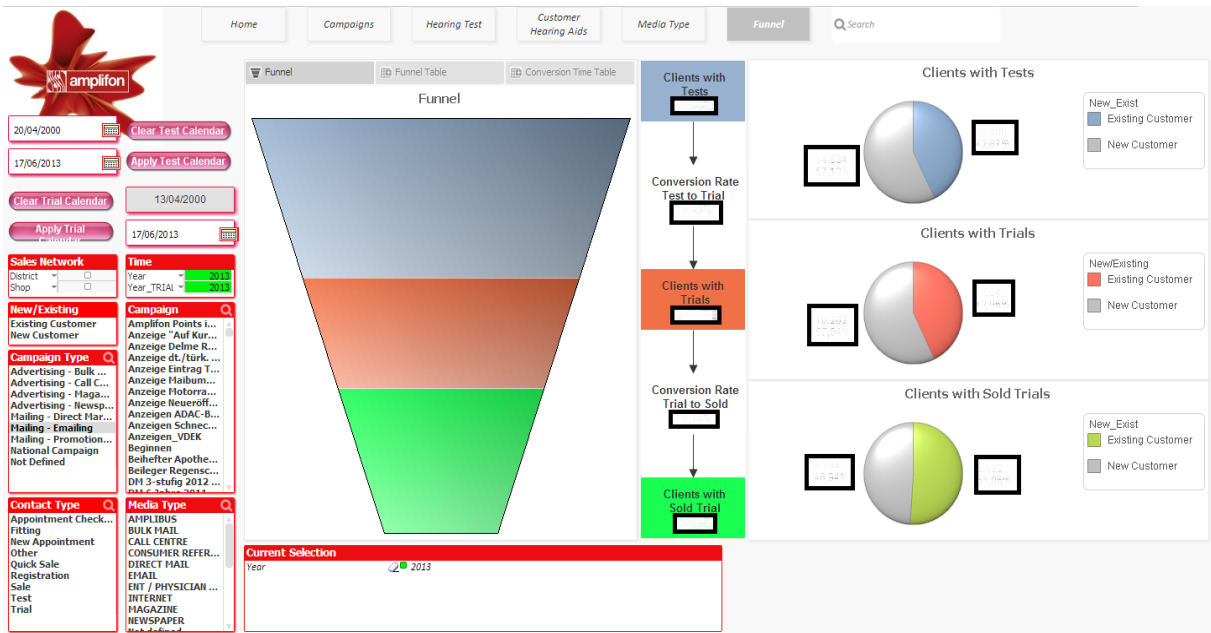


KPI visualizzati in questo report sono descritti al paragrafo 4.1.5 nella sezione relativa al report Media Type.

*Grafico a Torta:* distribuzione dei contatti per Media Type.

## Funnel e Conversion Rate

Questo è il report con la maggiore complessità di tutto il sistema sviluppato. La complessità dipende non solo dal lato tecnico dell'informazione calcolata (logica del funnel diversa dalla logica del sistema), ma anche dal lato Presentation per la quantità di informazione offerta. Si è cercato di utilizzare tecniche di *Optical View* per distribuire nel modo migliore possibile l'informazione.



In questa prima grafica vengono visualizzati solo grafici. L'utente può cambiare il grafico per passare alla visualizzazione della tabella relativa ai numeri del funnel e a quella relativa ai valori del Conversion Time, mostrati di seguito:

Funnel Table

Year	Clients with Test	Clients with Trial	Clients with Delivered Trials	Clients with Sold Trials
2013	13001	11209	7	12
<b>Total</b>	<b>13001</b>	<b>11209</b>	<b>7</b>	<b>12</b>
Amplifon Point...	1	0	0	0
Anzeige "Auf Kur...	0	0	0	0
Anzeige Delm...	0	0	0	4
Anzeige dt./türk...	44	0	0	2
Anzeige Eintr...	1	2	0	1
Anzeige Moto...	1	1	0	1
Anzeige Neue...	1	0	0	0
Anzeigen ADA...	1	0	0	0
Anzeigen Sch...	1	0	0	0
Beginnen	1	0	0	0
Beliefter Apo...	18	11	0	0
Belieger Rege...	1	0	0	0
DM Altona/Ot...	1	0	0	0
DM Batt.-Guts...	0	0	0	0
DM Christstoll...	4	0	0	0
DM Geburtsta...	200	47	1	1
DM Geburtsta...	1	0	1	28
DM LTZ 0,nix...	4	0	0	0
DM LTZ 0,nix...	1	0	0	3
DM LTZ 6 Jahr...	1	2	0	1
DM LTZ 6 Jahr...	2	2	0	1
DM LTZ Empfe...	1	0	0	0
DM LTZ Hörge...	1	0	0	0
DM LTZ Hörge...	20	0	0	0
DM LTZ Hörge...	1	0	0	0
DM LTZ Hörge...	1	0	0	1
DM LTZ Kinder...	1	0	0	0
DM LTZ Otool...	0	0	0	0

Conversion Time Table

Campaign	AVG Conv Time Test to Trial	MAX Conv Time Test to Trial	AVG Conv Time Test to Sale	MAX Conv Time Test to Sale
Amplifon Point...	1,00	1	-	-
Anzeige "Auf Kur...	0,00	0	35,50	78
Anzeige Delm...	0,00	0	-	-
Anzeige dt./türk...	1,75	7	54,00	56
Anzeige Eintr...	3,00	3	-	-
Anzeige Moto...	-	-	-	-
Anzeige Neue...	2,80	10	-	-
Anzeigen ADA...	0,00	0	-	-
Anzeigen Sch...	0,00	0	-	-
Anzeigen_VDEK...	-	-	-	-
Beginnen	229,00	567	1.125,00	1125
Beliefter Apo...	19,00	4	32,70	101
Belieger Rege...	0,00	0	50,00	43
DM 3-stufig 2...	0,00	0	-	-
DM 6 Jahre 2...	-	-	-	-
DM 6 Jahre W...	-	-	-	-
DM 6 Jahre W...	-	-	-	-
DM Altona/Ot...	-	-	-	-
DM Batt.-Guts...	0,00	0	-	-
DM Christstoll...	0,00	0	70,00	70
DM Frühlingsf...	-	-	-	-
DM Geburtsta...	75,00	700	85,00	760
DM Geburtsta...	8,00	700	91,00	91
DM Geburtsta...	0,00	0	110,00	115
DM LTZ 0,nix...	70,00	74	75,00	85
DM LTZ 6 Jahr...	7,67	10	51,00	41
DM LTZ 6 Jahr...	6,00	10	32,14	71
DM LTZ Empfe...	3,00	9	51,00	51
DM LTZ Hörge...	21,00	239	-	-
DM LTZ Hörge...	170,00	0	-	-
DM LTZ Hörge...	0,00	0	-	-
DM LTZ Hörge...	14,00	2038	39,00	58
DM LTZ Kinder...	0,00	0	-	-
DM LTZ Otool...	0,00	0	-	-

KPI visualizzati in questo report sono descritti al paragrafo 4.1.5 nella sezione relativa al report Funnel.

### *Grafici:*

- *Funnel:* mostra l'imbuto relativo alla logica di funnel, con i Clienti che hanno avuto un test (in Blu), di questi chi ha avuto una trial (in rosso), e per finire, di questi chi ha avuto una Trial Sold (considerato evento di vendita, in verde)
- *Riquadri centrali:* mostrano i valori del funnel in numeri e, fra due riquadri, i conversion rate.
- *Grafici a Torta:* mostrano valori del funnel in base ai New Customer ed Existing Customer.

### Osservazioni:

Per generare questo report abbiamo effettuato diverse comunicazioni preliminari con gli utenti di test, al fine di avere l'approvazione dell'informazione generata. Nonostante questo sono state effettuate diverse modifiche nella fase di post rilascio. Alcune di queste modifiche sono risultate talmente invasive, ma necessarie, da generare un mini progetto con over budget successivo (vedi Capitolo 5).

### 4.3.6 Technical Documentation

Questa fase del progetto ha rappresentato la chiusura delle attività di sviluppo. La documentazione tecnica rappresenta una delle attività di maggior valore per l'utente, in quanto rappresenta il momento in cui vengono preparati gli utenti di test all'attività formativa successiva.

La documentazione tecnica è un vero e proprio libretto di istruzioni per gli utenti, in particolare per quelli del reparto IT, che avranno il compito di mantenere il sistema e rispondere, ove possibile, alle richieste successive degli utenti finali.

La documentazione tecnica segue uno specifico schema di costruzione, basato sulle procedure standard di stesura dei documenti aziendali di Iconsulting. Infatti la suddivisione degli argomenti e la stesura del codice è fatta valutando l'importanza delle nozioni che si vuole trasmettere, evitando di confondere il lettore fra righe di codice inutile.

È stata posta grande attenzione, durante lo sviluppo, nella generazione di un codice che sia ben commentato e comprensibile, in modo tale che sia facilmente leggibile e raggiungibile per le modifiche successive.

*In questo elaborato non è stato inserito codice di programmazione ma solo la logica, in quanto non ho ritenuto importante l'inclusione del medesimo codice.*

Basandoci su tecniche di best practices, si è deciso di organizzare le pagine di codice QlikView sfruttando dei Tab (fogli elettronici) distinti per argomenti. In questo modo abbiamo creato una suddivisione semantica e "commentabile" del codice. Infatti tutti i file di Script (ETL01, ETL02 e Cloud) presentano dei Tab in comune, che svolgono le stesse operazioni logiche, contestualizzate nel file in cui si trovano (es. di tab comuni: Info, Renaming, Controll). Tutto questo ha facilitato la stesura della documentazione, che in questo modo si è concentrata sugli aspetti significativi del codice.

Nell'immagine sottostante si può notare la sequenza dei tab per i singoli file.



Da questo si è partiti per sviluppare in modo corretto e comprensibile la documentazione tecnica.

### **4.3.7 Formation**

La Formazione è stata l'attività conclusiva del progetto, in cui gli utenti sono stati istruiti sulle funzionalità e le proprietà del nuovo sistema sviluppato.

L'attività è stata svolta ad Amburgo, nella sede centrale di Amplifon DE.

Nei due giorni previsti per la formazione, abbiamo suddiviso le attività nel seguente modo:

- Introduzione e descrizione delle attività svolte (10% del tempo previsto)
- Formazione funzionale agli utenti finali (30% del tempo previsto)
- Formazione tecnica agli utenti dell'IT (60% del tempo previsto)

#### **Introduzione e descrizione delle attività**

È stata fatta una rapida introduzione dell'azienda e delle metodologie agile, con focus sulle logiche direttamente applicate nel progetto.

#### **Formazione Funzionale**

La formazione è stata effettuata direttamente dagli utenti di Test, che hanno acquisito padronanza del sistema nel corso dello sviluppo. Ovviamente gli utenti di test sono stati assistiti dagli sviluppatori.

La formazione è avvenuta direttamente sul sistema, al quale tutti gli utenti sono stati precedentemente abilitati. Partendo da una rapida descrizione delle funzionalità del sistema, si è passati, report per report, alla navigazione e al test pratico sullo stesso.

Si è scelto di dare la possibilità di interagire, per ogni report, con il sistema, in modo da permettere agli utenti di provare immediatamente le funzionalità del report descritto, senza attendere la fine della descrizione generale.

Alla fine della formazione funzionale, sono stati raccolti i dubbi e le richieste sollevate dagli utenti, in modo da dare la possibilità agli utenti IT di integrarle durante la formazione tecnica.

## **Formazione Tecnica**

Questa è stata la principale attività del percorso formativo. Durante questa fase ho effettuato direttamente la formazione al responsabile dell'IT, che è stato il principale interlocutore. L'intento è stato quello di insegnare tutti gli strumenti possibili al sistemista, in modo tale che quest'ultimo acquisisca una padronanza e competenza del sistema tale da poter rispondere a tutte le richieste.

In questo modo Iconsulting (lo sviluppatore) accompagna Amplifon DE (l'utilizzatore finale) verso l'autosufficienza, uscendo gradualmente di scena.

La formazione è avvenuta su tre fasi:

- *Fase descrittiva*: descrizione tecnica del sistema e formazione dell'utente per l'utilizzo della documentazione tecnica.
- *Fase di test*: sono stati sottoposti dei piccoli esercizi pratici per l'utilizzo del sistema.
- *Fase pratica*: sono stati risolti tutti i quesiti e sviluppate le modifiche richieste dagli utenti durante la fase di formazione.

Al termine di questa fase si possono dichiarare concluse tutte le attività relative al progetto pianificato.

# Capitolo 5. Evoluzioni (CR Sales)

Durante lo sviluppo sono emerse una serie di inconsistenze, logiche e funzionali, erroneamente valutate o di cui, i sistemisti Amplifon, ne ignoravano l'esistenza.

Visto che le variazioni avevano un impatto notevole sul tempo di sviluppo, e di conseguenza sul budget previsto inizialmente, è stata sviluppata una seconda fase del progetto, comunemente detta **CR** (change requirement).

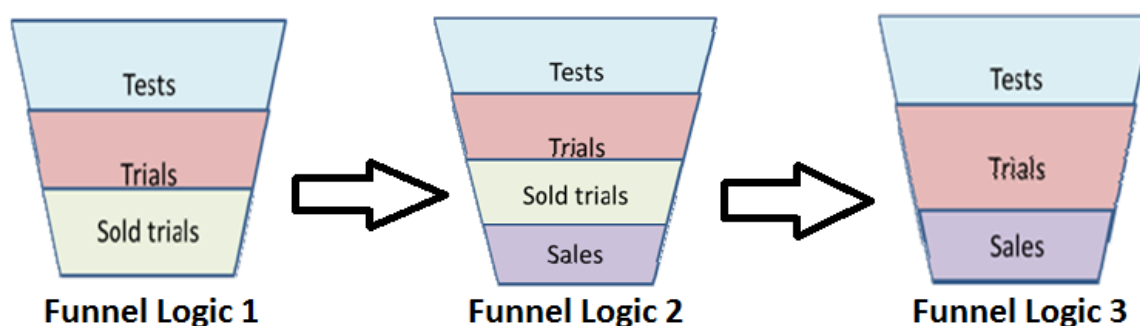
Il progettino ha preso il nome di *CR Sales*, in quanto riguarda le vendite nel funnel.

## 5.1 Evoluzioni Funzionali

Dopo la condivisione dei requisiti, e l'approvazione dell'intera serie di attività, abbiamo cominciato la fase di progettazione e sviluppo del sistema, che ha attraversato diverse difficoltà nella parte di sviluppo della logica del funnel.

Lo strumento utilizzato, QlikView, ha mostrato le sue limitazioni nell'implementazione delle logiche di questo tipo. Infatti, per risolvere le difficoltà incontrate, abbiamo dovuto creare una replica della tabella dei fatti centrale, solo per le Trial, riuscendo così a rispondere alla logica del funnel richiesta.

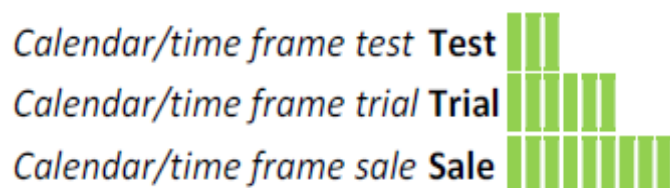
Nella fase iniziale si credeva che il documento di *Sold Trials* fosse sufficiente ad individuare l'evento di vendita (uno degli scopi del funnel), invece ci si è resi conto che non è stato così.





Come si nota dalla sequenza, la logica iniziale non era completa, e visto che l'intento era quello di tracciare il contatto di una persona dal momento del test, al trial fino a concludere con la vendita, si è dovuto modificare la logica del sistema.

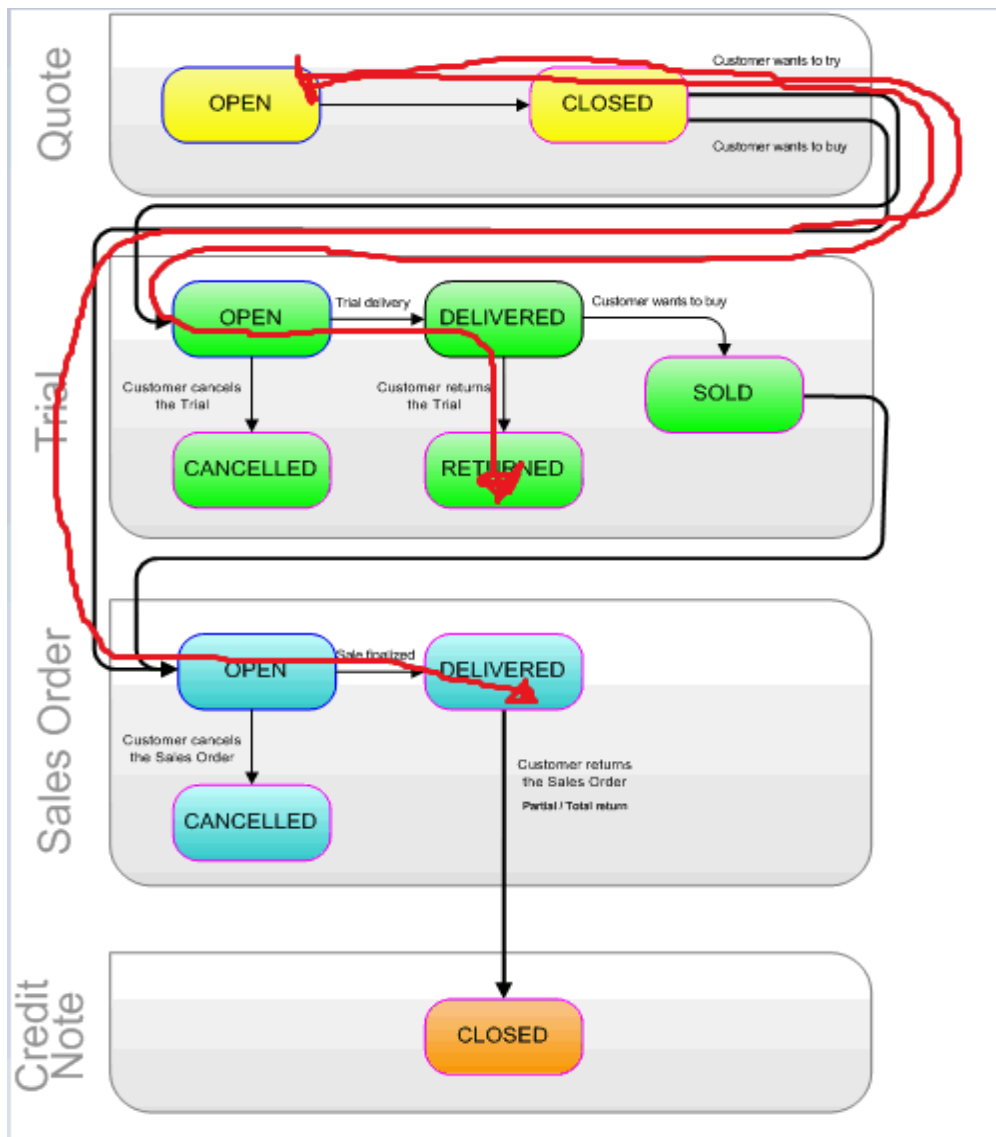
Oltre a tracciare gli eventi in forma completa, si dà all'utente la possibilità di poter tracciare le variazioni con tempi variabili e selezionabili dinamicamente per i singoli eventi. L'esempio sottostante mostra la logica teorica da utilizzare per la selezione degli intervalli temporali.



Così l'utente è in grado di effettuare ricerche con una personalizzazione maggiore, vincolato però dal fatto che il periodo del livello più alto (test) debba essere sempre maggiore o uguale del livello sottostante (come mostrato sopra).

Inoltre anche nel calcolo dei *conversion time* si è deciso di includere i tempi di cambio fra TEST-TRIAL, TRIAL-SALES, TEST-SALES.

Alla fine dell'intero processo si è reso possibile il rintracciamento del flusso delle informazioni del singolo cliente, partendo dal primo evento strategico (il TEST), passando per il secondo (TRIAL), fino a raggiungere il fattore di maggior valore (SALES) escludendo eventuali ritorni (CREDITE NOTES).



La figura mostra chiaramente l'intero percorso che è stato tracciato.

Questo può essere fatto per qualsiasi dimensione coinvolta. Ovviamente per il reparto marketing risulta strategico nell'analisi delle campagne promozionali, infatti è possibile valutare l'efficacia di una campagna in base al numero di utenti che sono stati coinvolti e al numero di coloro che hanno portato al termine il ciclo di vita con l'acquisto.

## 5.2 Analisi e Sviluppo CR Sales

Anche per questa piccola evoluzione progettuale è stata fatta una stima delle attività applicando il modello di progettazione incrementale.

Di seguito vengono rappresentate le stime e le iterazioni sulle attività di progetto:

Funzionalità	Week 1	Week 2
<b>Solution Analysis</b>		
<b>Modeling DataMart</b>		
<b>Staging Area</b>		
<b>Metadata Definition</b>		
<b>Developing UAT Reports</b>		★
<b>Technical Documentation</b>		
<b>Formation</b>		★

★ Delivery Reports & UAT.

★ Documentation Delivery.



Revisione step precedenti.

Revisione degli step successivi in base ai precedenti.

In questo caso le prime fasi sono state più rapide in quanto prevedevano piccole o, in alcuni casi, nessuna variazione al modello precedente.

Il tempo complessivo del progetto è stato di due settimane, anche se il tempo di sviluppo si era notevolmente ridotto, in quanto si è semplicemente estesa la logica del funnel precedentemente applicata.

## **Solution Analysis**

L'analisi della soluzione ha tenuto conto di una serie di limitazioni dello strumento, e si è scontrata con le diverse opinioni delle figure coinvolte, infatti il concetto di vendita assume significati particolari, da reparto a reparto, e l'evento di vendita viene tracciato con criteri diversi nel sistema informativo di Amplifon.

Per quanto concerne i test in fase preliminare, abbiamo effettuato una serie di controlli sul sistema, per capire se le informazioni presenti siano consistenti e per verificare che il modello dei dati costruito non necessiti di variazioni per risolvere la CR.

Nella fase di Test, siamo riusciti a rispondere e a chiarire dei concetti che non erano molto chiari al reparto IT di Amplifon, ad esempio casi di persone che hanno effettuato dei test e poi direttamente l'acquisto.

## **Modeling Datamart**

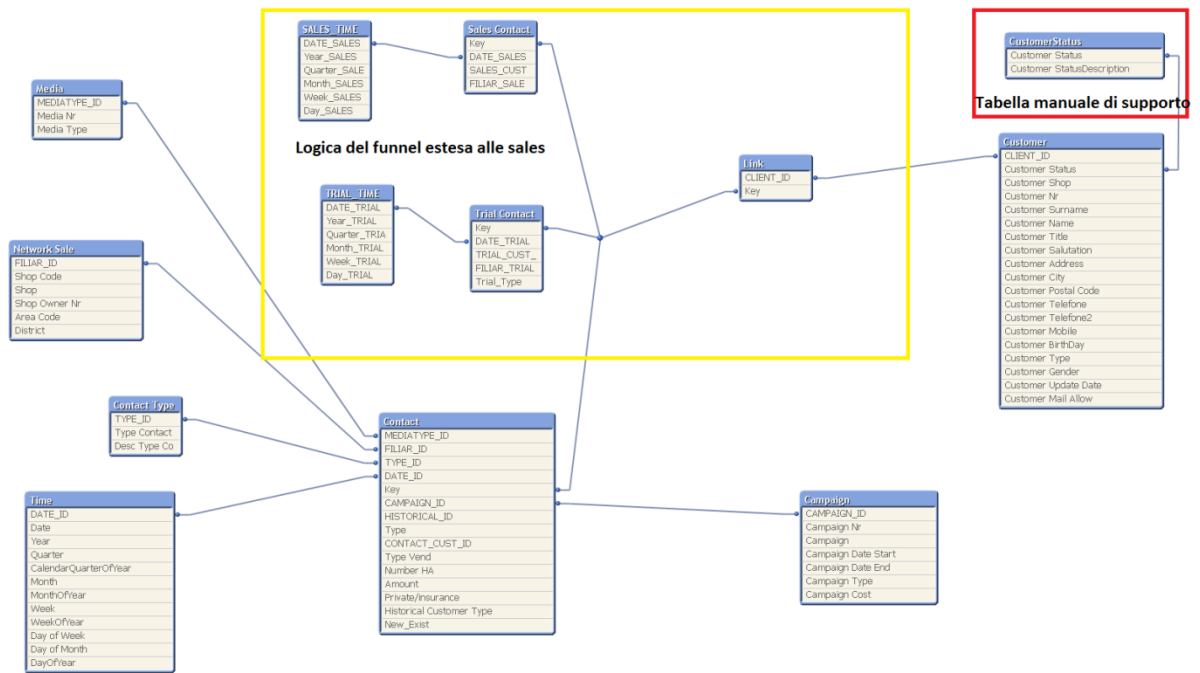
Questa fase non ha subito variazioni rispetto a quella del precedente progetto, tutte le tabelle che sono state importate contenevano già le informazioni necessarie per rispondere alle nuove richieste.

## **Staging Area**

Come per la precedente fase, anche in questo caso non abbiamo avuto necessità di effettuare modifiche a livello di Staging Area. I file di ETL01 e ETL02 sono rimasti sostanzialmente invariati.

## **Metadata Definition**

Questa è stata la fase principale di questo progetto. Abbiamo riorganizzato il file di Cloud per generare, da una parte le tabelle manuali, che forniscono informazioni utili da visualizzare per l'utente. Dall'altra abbiamo dovuto estendere la logica del funnel, utilizzata per le trial, alle sales. Questo ha comportato le seguenti variazioni della nuvola.

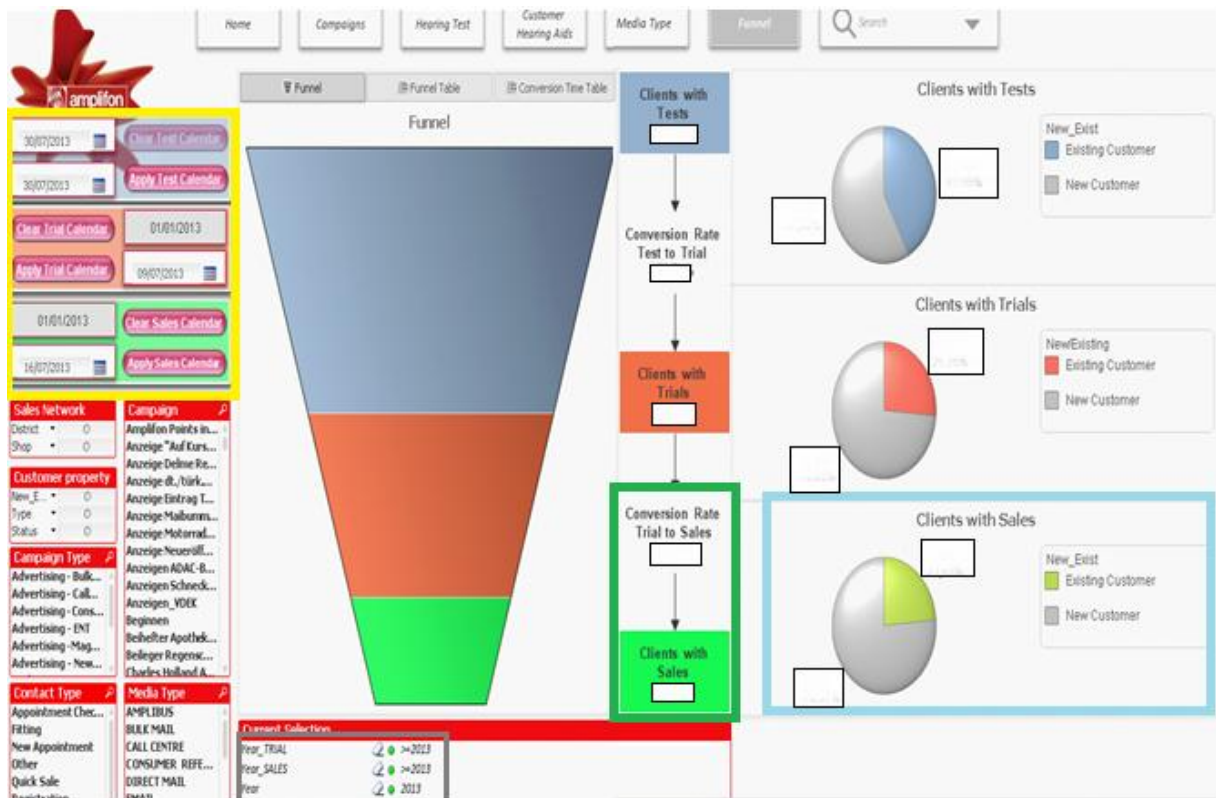


Per implementare la nuova nuvola, abbiamo dovuto effettuare una serie di Regression test di confronto fra il precedente sistema e quello successivo. La logica ha previsto la generazione di una fact table di supporto, specializzata sulle sales, che ha lo scopo di considerare gli utenti di test (selezionati dalla CONTACT), che hanno effettuato una trial (selezionati dalla TrialContact), e che hanno effettuato anche una sales (SalesContact). Allo stesso tempo si è data la possibilità di scegliere le finestre temporali parzialmente indipendenti fra loro (vedi paragrafo 5.1).

### Developing UAT Reports

In questa fase si è passati alla revisione diretta del report Funnel. Per gli altri report è stata fatto soltanto il controllo per verificare che il nuovo modello non abbia apportato modifiche indirette sugli altri.

Il report del funnel è stato rivisto nella sua totalità, aggiornando i grafici e le tabelle alla nuova logica di calcolo.



Le variazioni principali riguardano:

- La parte relativa ai filtraggi che si è arricchita del calendario per le sales. Adesso si ha la possibilità di selezionare un periodo maggiore uguale del periodo trial, e così a ritroso. I calendari sono ora dello stesso colore del contatto, così da avere una maggiore immediatezza nella comprensione del calendario;
- La parte relativa al riquadro per le sales ed ai conversion rate è stata aggiornata al valore corretto;
- Il grafico a torta misura ora i clienti che hanno avuto una vendita, in base alla dimensione New/Existing Customer.



# Capitolo 6. Conclusioni

In questo ultimo capitolo tratteremo dei risultati ottenuti da:

- Amplifon DE, per la capacità di effettuare ricerche in modo efficace ed efficiente.
- Iconsulting, per le competenze di progettazione e di sviluppo acquisite.

Nella descrizione dei risultati faremo esempi concreti di come il sistema e le tecniche di progettazione siano cambiate prima e dopo il sistema sviluppato.

## 6.1.1 Miglioramenti della soluzione sviluppata

Grazie alla nuova soluzione sviluppata, il reparto marketing di Amplifon DE ha migliorato notevolmente la capacità di fare ricerche mirate o generiche sull'efficacia delle campagne pubblicitarie, o sulla tipologia di mezzo utilizzato, o sulle aree maggiormente coinvolte e non, e così via.

Tutto questo è stato reso possibile grazie ad un sistema agile ed efficiente, in grado di fornire risultati in tempi brevissimi, quasi istantanei, processando anche milioni di informazioni.

Un esempio di come vengono visualizzate le informazioni fra i due sistemi è il seguente:

Vecchio sistema di CRM utilizzato:



Persönliche Daten   Krankenkasse   Marketing   Tests   Historie

**Kundendaten**

Kundennr. 13101105   Adresszeile Ilsestr. 6  
 Titel   Adresszeile 2  
 Anrede   Adresszeile 3  
 Vorname   Postleitzahl 10318  
 2. Vorname   Ort Berlin  
 Nachname Tlusteck   Land Germany  
 Geb.-Datum 28.08.1924   Alter 88   Vorw. - Tel. 1 030  
 weiblich    männlich   Telefon 2  
 Sprache German   Handy  
 eMail   Fax

**Kontaktperson**

Name   Kundentyp CUSTOMER   03.06.2010  
 Anschrift   Kundenstatus Aktiv  
 Telefon   Kategorie Kunde  
 Beziehung   Versicherungsart gesetzlich  
 Mitarbeiter  
 Medien Erstkontakt Zeitschrift / Magazin  
 Kampagne Erstkontakt Beihfetter Apotheken-Umschau 06/2013

**Kundeninfo**   Notizen

**Krankenkassen Daten**  
 Versicherung: 101575519 - Techniker Krankenkasse   Versichertennummer: 3113836297

**Termin Informationen**  
 Nächster Termin:   Letzter Termin: 02.07.2013

Links		Rechts	
Hörgerät:	Siemens A.T. GmbH INTUIS LIFE VU53298	Hörgerät:	Siemens A.T. GmbH INTUIS LIFE VU53442
Empfänger/Hörer:		Empfänger/Hörer:	
Fernbedienung:		Fernbedienung:	
Batterie:		Batterie:	
Lieferdatum:	03.06.2010	Lieferdatum:	03.06.2010
Garantieende:	03.06.2011	Garantieende:	03.06.2011

## Nuovo sistema QlikView:

Main   Campaign   Hearing Test   **Hearing Aids**   Media Type   Funnel

me   Campaigns   Hearing Test   Customer Hearing Aids   Media Type   Funnel   Search

Customer HA by Shop

Shop	Customer BirthDay	Customer Nr	Customer Type	Title	Salutation	Name	Surname	Gender	Street	Postal Code	City	Latest Sales Order	Second Last Sales Order	Third Last Sales Or...	Mail Allowed	No Call Flag	Author. Provi...
Berlin-Friedrichsf...	28/08/1924	13101105	CL	ND			Tlusteck	F	Ilsestr. 6	10318	Berlin	03/06/2010	-	-	N	N	Y

Come si nota chiaramente, la nuova visualizzazione è molto più immediata e leggibile, in quanto racchiude in una sola linea di tabella tutte le informazioni che sono strettamente necessarie al reparto marketing, trascurando tutto il resto.

Il sistema sviluppato, da ora la possibilità di effettuare ricerche generiche e particolari su:

- Campagne Promozionali
- Tipologia di Media utilizzato (email, tv, ecc.)
- Funnel
- Tipologie di contatto.

Tutte queste ricerche venivano fatte manualmente, col sistema precedente, ed inoltre risultavano poco affidabili, in quanto non tenevano conto di fattori da trascurare, vedi

il calcolo per le sales che andava a considerare direttamente un contatto di tipo sales generico e non specifico (Tipo Hearing Aids).

Per questo motivo il nuovo sistema è stato di grande valore per il cliente, che ha potuto apprezzare, non soltanto la capacità di raggiungere gli obiettivi prefissati, nei tempi e modi stabiliti, ma anche la possibilità di entrare in contatto maggiormente con la logica del proprio sistema IT, del quale non aveva molta confidenza e continuava ad usarlo facendo considerazioni corrette a livello logico, ma errate a livello sistemistico (ad esempio considerando un contatto Trial Sold come evento di vendita, mentre invece il contatto poteva assumere significati diversi).

In conclusione, da parte di Amplifon, c'è stato grande apprezzamento per l'impegno e la serietà con cui si è portato a termine il progetto.

Per questo motivo dopo questo progetto sono nate diverse opportunità di collaborazione fra l'azienda sviluppatrice Iconconsulting e il cliente Amplifon.

### **6.1.2 Competenze acquisite**

Dal punto di vista interno (di ICONCONSULTING), si è portato avanti un discorso di crescita di competenze nell'ambito della programmazione di QlikView. Infatti dopo una serie di incontri e corsi fatti con esperti del software, si è riusciti a delineare e a costruire un documento ben formato, nel quale sono state raccolte tutte una serie di tecniche e best practices di progettazione.

Prima, le competenze interne condivise erano limitate alla sola logica di costruzione base di un progetto, senza applicare approcci particolari di sviluppo dello stesso, generando un codice poco manutenibile. Oggi l'azienda può contare su di una documentazione ben precisa, che descrive passo dopo passo le procedure da seguire per impostare e sviluppare un progetto in maniera efficiente. Le seguenti procedure, descritte in parte nel paragrafo 3.23.2, se seguite correttamente permettono di raggiungere in modo rapido ed efficace la costruzione di un modello, e allo stesso tempo facilitano le azioni di manutenzione successive.

Inoltre, la documentazione del progetto Amplifon De, è stata scritta in modo tale da poter essere utilizzata come caso di studio per progetti QlikView successivi, condivisa internamente e riassunta in un documento introduttivo.

Concludendo possiamo affermare che il progetto ha arricchito da una parte l'azienda, in termini di competenze ed esperienza, dall'altra mi ha permesso di entrare a conoscenza di un mondo nuovo, dal quale spero di aver tratto i migliori benefici, impegnandomi ad apprendere tutto quello che mi è stato proposto.

Ringrazio ICONSULTING e i ragazzi che mi hanno seguito perché mi hanno insegnato non solo strumenti, tecniche e linguaggi, ma soprattutto quello spirito di ICO, che ti unisce e ti permette di superare le difficoltà col sorriso e con l'aiuto incondizionato di ognuno. Grazie ragazzi.

## Bibliografia

“The Role of the Multidimensional Database in a Data Warehousing Solution”. Arbor Software, 1995.

Scott Brown. “Try Slice and dice”. Computing Canada, Vol. 21, No. 22, Ottobre 1995.

Luca Cabibbo e Riccardo Torlone. “A Logical Approach to Multidimensional Databases”. Università di Roma Tre, Rapporto Interno.

Kevin Fogarty. “Data mining”. Network world, Vol. 11, No. 23, Giugno 1994.

Jim Gray e Andreas Reuter. “Transaction processing : concepts and techniques”. Morgan Kaufmann, 1993. ISBN : 1-55860-190-2.

William H. Inmon. “Building the Data Warehouse”. John Wiley & Sons, 1992.

Ralph Kimball. “The Data Warehouse Toolkit”. John Wiley & Sons, 1996.

Ralph Kimball, Laura Reeves, Margy Ross e Warren Thornthwaite. “The Data Warehouse Lifecycle Toolkit : expert methods for designing, developing, and deploying data warehouse”. John Wiley & Sons, 1998.

“The Case for Relational OLAP”. Microstrategy Incorporated, 1995.

“Data Warehousing, Data Modelling & Design”. Microstrategy Education, 1996.

“Fundamentals of DSS Agent & Architect”. Microstrategy Education, 1998.

"The Art of Agile Development " by James Shore, Chromatic. O'Reilly 2008.

"The Agile development" Slides del corso interno in ICONSULTING.

QlikView. Disponibile al sito: <http://www.qlikview.com/>

“QlikView 11 for Developers” by Miguel Garcia and Barry Harmsen. Packt Publishing.

QlikView Comunity. Disponibile al sito: <http://community.qlikview.com/welcome>

QlikView Slides: disponibili all'interno della sezione Community di QlikView

- QlikView Optimization Best Practices,
- QlikView Reference Manual,
- Developing on QlikView,
- QlikView Server Reference Manual

“GNU General Public License”. Wikipedia, 2013. Disponibile al sito:

<http://it.wikipedia.org/>

Amplifon: Informazioni ottenute dalla rete Intranet aziendale, accessibile solo dall'interno dei loro sistemi.