

UNIVERSITÀ DEGLI STUDI DEL SANNIO

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

Un framework per l'evoluzione dei sistemi informativi aziendali: implementazione del cambiamento

Relatori:

Ch.mo Prof.ssa Lerina Aversano

Ch.mo Prof.ssa Maria Tortorella

Candidato:

Dott. Francesco Cioffi

matricola: 393 / 000090

ANNO ACCADEMICO 2009 / 2010

*ogni volta che il termine informatica viene accostato
a una seconda parola si crea una scienza (nuova?)*

(Maurizio Pighin, Anna Marzona, "Sistemi informativi aziendali")

*Alla mia famiglia
che mi ha sostenuto durante tutti questi anni.*

*Alla mia fidanzata, Silvia, per il supporto,
senza cui questo lavoro sarebbe stato senz'altro meno piacevole.*

*Si ringraziano di vero cuore le relatrici per il supporto e per la disponibilità
e per il clima di serena collaborazione che hanno saputo instaurare
rendendo gradevole e mai noioso il lavoro svolto.*

*Si ringrazia il Dott. Mario Rosa per la collaborazione,
il quale ha reso possibile il confronto con un contesto reale.*

*In bocca al lupo a Rosa,
con cui ho condiviso gran parte di questo lavoro.*

Indice generale

Premessa.....	8
Capitolo 1	
Evoluzione dei sistemi informativi aziendali.....	12
1.1 Che cos'è un SIA.....	12
1.2 Cenni storici.....	15
1.3 Il mercato dei sistemi informativi, soluzioni proprietarie e libere.....	18
1.3.1 SAP: una soluzione commerciale.....	20
1.3.2 Soluzioni open source.....	22
1.3.2.1 Openbravo.....	22
1.3.2.2 Compiere.....	23
1.4 I sistemi informativi per le PMI.....	25
1.5 Evoluzione dei sistemi informativi: eventi interni ed esterni.....	27
Capitolo 2	
Stato dell'arte.....	30
2.1 Introduzione.....	30
2.2 Architecture-Driven Modernization.....	31
2.3 Model-Driven Software Modernization.....	36
2.4 Un framework basato sulla metodologia MDE di Alexandre Cláudio de Almeida et al.....	38
2.5 Un framework basato sulla disciplina ESE di Robert S. Swarz e Joseph K. DeRosa.....	41

2.6 Un modello per la manutenzione di un ERP di Celeste See Pui Ng. et al.	44
2.7 Un framework per la gestione del cambiamento di Adel M. Aladwani.....	45
2.8 Le metodologie a confronto.....	49
2.9 Altre metodologie.....	55
2.9.1 Evoluzione dinamica.....	55
2.9.2 Un approccio verso la qualità.....	56

Capitolo 3

Un framework per la modernizzazione dei SIA di PMI italiane.....	58
3.1 Obiettivi.....	58
3.2 Descrizione del framework.....	59
3.3 Le fasi di implementazione del cambiamento.....	70
3.3.1 Integrazione del sistema target.....	71
3.3.1.1 Sviluppo / modifica delle componenti software mancanti.....	72
3.3.2 Sostituzione del sistema.....	75
3.3.2.1 Suddivisione in categorie funzionali / organizzative.....	76
3.3.2.2 Pianificazione della sostituzione del sistema.....	78
3.3.2.3 Definizione ed applicazione delle strategie di gestione del cambiamento.....	79
3.3.2.4 Training su piattaforma di test.....	81
3.3.2.5 Migrazione dei dati ed evoluzione delle procedure.....	82
3.3.2.6 Verifica e supporto.....	84
3.3.2.7 Iterazione successiva e sviluppo incrementale.....	85
3.3.3 Verifica della modernizzazione.....	86

3.3.3.1 Verifica finale.....	86
3.3.3.2 Dismissione vecchio sistema.....	87
3.4 Confronto con le metodologie definite in letteratura.....	88
Capitolo 4	
Caso di studio.....	90
4.1 Contesto.....	90
4.2 Architettura del SIA attuale.....	92
4.3 Applicazione del framework.....	98
4.3.1 Analisi e progettazione.....	98
4.3.2 Implementazione del cambiamento.....	99
4.3.2.1 Integrazione del sistema target.....	100
4.3.2.2 Sostituzione del sistema.....	101
4.3.2.3 Verifica della modernizzazione.....	102
Conclusioni e sviluppi futuri.....	103
Appendice A	
Procedure aziendali della società FRTB.....	106
Ciclo passivo.....	106
Ciclo attivo.....	109
Rifatturazione.....	110
Rivendite fidelizzate.....	111
Alcune precisazioni.....	112

Appendice B

Sistema gestionale target della società FRTB.....	113
Architettura.....	113
Sincronizzazione.....	115

Premessa

Scopo del lavoro di tesi è la definizione di un framework per l'evoluzione di sistemi informativi aziendali. Per evoluzione si intende il processo di modernizzazione di quei sistemi software che sono da supporto alle attività aziendali.

La prospettiva da cui si analizza la problematica è quella delle PMI, realtà che riveste una percentuale considerevole dell'occupazione in Italia.

Le motivazioni che spingono un'impresa ad avviare un processo di evoluzione possono essere varie ma sempre riconducibili ad eventi sia interni che esterni all'impresa. Si può avviare un processo di evoluzione per mantenere la propria competitività sul mercato e per migliorare i processi produttivi in quanto l'innovazione tecnologica ne rende possibile il miglioramento. D'altra parte l'evoluzione può essere necessaria per migrare codice legacy, ossia codice sorgente correlato ad un sistema non più supportato ovvero non più in produzione.

Il processo di evoluzione è piuttosto complesso poiché va a modificare ciò

che sono le procedure in atto nel sistema azienda ovvero i sistemi software utilizzati. Inoltre il passaggio vero e proprio, se non ben programmato, può portare a disagi quali tempi di inattività dell'impresa. Quindi è necessario pianificare e programmare tale processo in modo accurato così da evitare disagi ma anche al fine di massimizzare il miglioramento ottenuto con la migrazione verso il sistema target.

Il presente lavoro di tesi è organizzato come segue.

Nel primo capitolo viene tracciata una panoramica sui sistemi informativi aziendali. Da alcuni cenni storici su come questi si sono evoluti nel tempo, grazie anche all'innovazione tecnologica avvenuta in ambito informatico, ad un'illustrazione di quelli attualmente presenti sul mercato, sia in ambito open source che commerciale. Particolare attenzione è prestata ai sistemi informativi aziendali delle PMI. In ultimo ci si sofferma sulle ragioni che inducono all'evoluzione di tali sistemi.

Nel secondo capitolo si studia lo stato dell'arte relativo ai framework di modernizzazione che sono stati definiti in letteratura e alle tecniche di supporto al processo stesso. I modelli analizzati si riferiscono anche a processi di trasformazione del software in generale, quindi vengono raffrontati con la realtà di interesse sempre con particolare riguardo a quella delle PMI.

Nel terzo capitolo viene presentato il framework definito in questo lavoro di tesi per l'evoluzione dei sistemi informativi aziendali, sviluppato e concepito in relazione alle esigenze delle PMI. Per lo sviluppo del framework si attinge ai

modelli descritti nel capitolo precedente rapportando il tutto alla realtà di interesse.

Nell'ultimo capitolo si descrive un caso di studio reale, ossia la realtà di un'azienda che si trova ad evolvere il proprio sistema informativo. Il caso viene esaminato in relazione alla metodologia sviluppata applicando, dunque, il framework definito nel capitolo precedente. Ne sono analizzate le motivazioni che hanno contribuito all'avvio del processo di evoluzione e le scelte effettuate, relative sia al sistema target sia alle fasi del processo stesso.

Il caso di studio è effettuato presso l'azienda Fabbriche Riunite Torrone di Benevento.

Il lavoro di tesi è realizzato in collaborazione con la Dott.ssa Rosa Principe, laureanda nella stessa università. Si troveranno di seguito riferimenti al Suo lavoro del quale si consiglia la lettura per una comprensione completa ed una trattazione esaustiva dell'argomento.

Capitolo 1

Evoluzione dei sistemi informativi aziendali

Una panoramica sui sistemi informativi aziendali, cenni storici ed il mercato attuale. Si analizzano le problematiche delle PMI riguardo l'adozione di tali sistemi e la loro evoluzione. L'obiettivo è di introdurre il lettore all'argomento.

1.1 Che cos'è un SIA

Un sistema informativo è l'insieme delle procedure e delle infrastrutture che definiscono e supportano il fluire delle informazioni all'interno di una struttura organizzativa.

I SIA (Sistema Informativo Aziendale) hanno come obiettivo finale la distribuzione di informazioni alle persone che operano all'interno dell'azienda

nel momento in cui l'informazione è necessaria. Il funzionamento del SIA prevede l'intervento di persone che interagiscono con le procedure di elaborazione.

Il SIA è quindi, l'insieme dei dati, delle procedure, dei modelli organizzativi e dei mezzi adottati per utilizzare l'informatica all'interno dell'azienda[1].

Diffondere la conoscenza all'interno dell'azienda è un compito piuttosto complesso in quanto le persone che vi operano hanno necessità di conoscenza dei fenomeni aziendali molto diverse, con esigenze che si differenziano in base al livello di astrazione, tempestività e livello di copertura. Il sistema informativo deve garantire che l'informazione sia accessibile nelle forme e nei tempi opportuni per ciascun operatore. Inoltre deve assicurare la correttezza del flusso delle informazioni.



Figura 1: piramide di Anthony

Una rappresentazione efficace della classificazione dell'esigenza

informativa è la piramide di Anthony¹ rappresentata in Figura 1: man mano che si sale nel livello decisionale la necessità di molte informazioni dettagliate decresce, mentre cresce l'esigenza di avere poche informazioni sintetiche di qualità che permettono l'adozione di decisioni strategiche corrette.

Le diverse esigenze informative, quindi, hanno portato nel tempo ad una separazione tra sistemi informativi orientati alle decisioni e sistemi informativi orientati al supporto dell'operatività e al controllo sistematico delle attività aziendali come schematizzato in Figura 2.

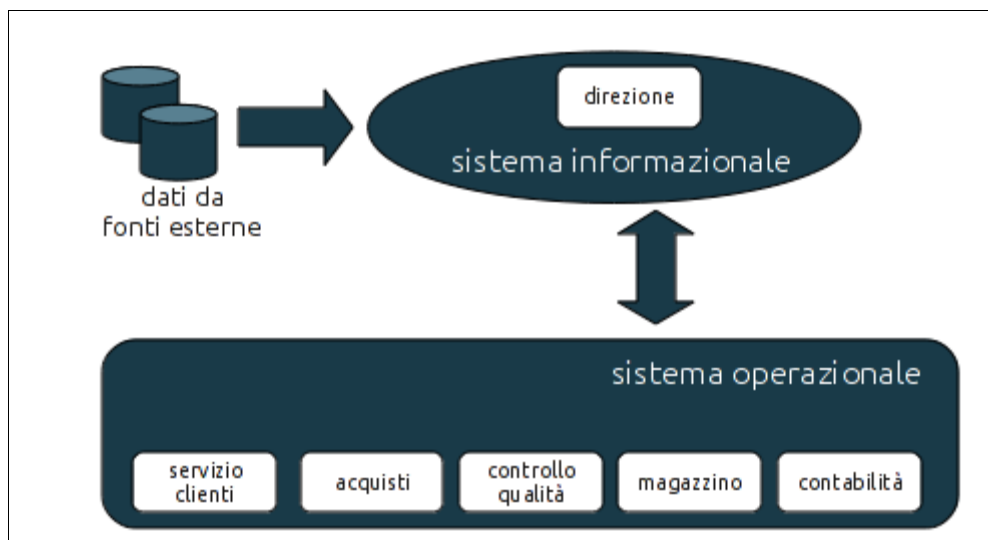


Figura 2: Relazione tra sistema operativo e sistema informazionale

I sistemi operazionali costituiscono l'infrastruttura informatica su cui si appoggia l'attività esecutiva, sono orientati quindi al trattamento delle attività quotidiane. Qualunque sia il settore in cui l'azienda opera il sistema operativo è composto da una base di dati su cui agiscono procedure di aggiornamento, interrogazione ed elaborazione. Copre tutte le aree funzionali e

¹ La piramide di Anthony (inglese: Anthony's pyramid, o, più raramente, Robert Anthony's pyramid) è un sistema di classificazione per l'organizzazione delle industrie, un modello gerarchico di comportamento organizzativo che ha influenzato il pensiero manageriale.

fornisce un supporto a gran parte dei processi operativi aziendali; si presenta nel suo complesso come un'entità estremamente articolata, tradizionalmente scomposta in sottosistemi di dimensioni minori rivolti alla gestione e al controllo di particolari aree o flussi esecutivi dell'azienda.

D'altra parte i sistemi informazionali supportano i processi decisionali seguendo i passaggi logici del decisore e dandogli la possibilità di avere visioni diversamente organizzate dei dati. L'esigenza informativa a livello direzionale richiede informazioni sui risultati rispetto agli obiettivi aziendali, strumenti per il confronto tra indicatori aziendali e parametri aziendali e strumenti che facilitano il processo decisionale tramite analisi interattive articolate sui dati. Contrariamente al sistema operativo l'accesso ai dati è di solito in sola lettura: gli utenti non eseguono operazioni CRUD (Create, Read, Update, Delete) sui dati, azioni che vengono effettuate alle funzioni di aggiornamento periodico sui dati.

1.2 Cenni storici

Già a partire dal 1970[2][3] si compiono i primi studi sull'applicazione dell'informatica ai sistemi informativi aziendali a supporto dei differenti processi gestionali, con lo scopo di coprire particolari esigenze o funzionalità. Tali sistemi sono separati, per lo più composti da applicazioni altamente personalizzate,

sviluppate da varie aziende di software con interfacce limitate e costose tra i diversi sistemi. Queste soluzioni sono poco flessibili e particolarmente difficili da integrare.

Il passo successivo avviene intorno al 1980, quando si sviluppano i MRP (Materials Requirements Planning), cioè moduli software dedicati alle esigenze di informatizzazione legate soprattutto alle problematiche di approvvigionamento dei materiali necessari alla produzione dell'azienda, con l'obiettivo di mantenere consistenti le informazioni nelle varie fasi di pianificazione dell'approvvigionamento, trasporto e consegna dei materiali. Le applicazioni sono interfacciate fra loro e spesso vengono sviluppate in tempi successivi da diverse aziende ma le informazioni sono parzialmente isolate, infatti ciascuna applicazione si appoggia su archivi specifici. La conoscenza dei dati è inclusa negli applicativi, quindi, qualora la struttura subisse dei cambiamenti si rende necessario modificare i programmi stessi. L'interfacciamento avviene tramite processi di elaborazione batch e off-line. Archivi differenti, e non integrati, contribuiscono a una grande ridondanza di dati e frequenti incoerenze fra gli stessi.

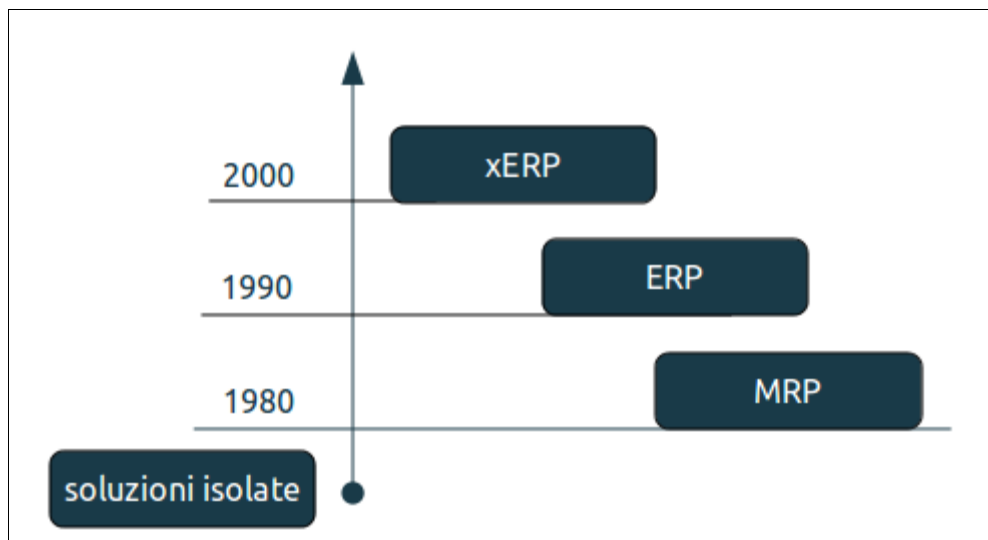


Figura 3: evoluzione storica dei SIA

Negli anni a seguire iniziano a comparire i primi sistemi di ERP (Enterprise Resource Planning) che oggi rappresentano uno standard de facto all'interno dei sistemi informativi aziendali. Negli ERP l'integrazione è nativa sia sotto il profilo dell'architettura informatica che della struttura logica. Mentre i MRP si preoccupavano prevalentemente di supportare i processi primari di produzione, gli ERP affiancano a questo la gestione dei processi di vendita e dei processi non direttamente legati alla produzione (contabilità, gestione del personale, ecc). Sono comunque tutti sistemi formati da centinaia di software che insieme si occupano di gestire la mole di dati necessaria alla gestione dei processi aziendali. Una differenza sostanziale sta nel fatto che i MRP utilizzano diverse porzioni di una complessiva base di dati, invece, gli ERP utilizzano un'unica base di dati condivisa. Ciò permette di realizzare una più completa integrazione in grado di ridurre tempi e costi di gestione dei dati, e consentendo lo sviluppo di una visione d'insieme più completa al fine di supportare meglio i processi decisionali delle funzioni aziendali preposte a questo.

Infine, negli ultimi anni hanno iniziato a diffondersi gli xERP (extended ERP) che sono caratterizzati dall'apertura del sistema ERP al mondo esterno per fornire informazioni aziendali ai partner dell'azienda, ovvero permettendo l'accesso controllato al sistema informativo aziendale da parte di fornitori e clienti. Di seguito alcune estensioni che si sono affermate nel tempo:

- CRM (Customer Relationship Management), per la gestione dei rapporti con la clientela;
- SFA (Sales Force Automation), per l'automazione dei processi di vendita;
- SCM (Supply Chain Management), per la gestione della catena di fornitura;
- PRM (Partner Relationship Management), per la gestione dei rapporti con i partner;
- strumenti per l'e-commerce.

In Figura 3 si riassume l'evoluzione storica illustrata.

1.3 Il mercato dei sistemi informativi, soluzioni proprietarie e libere

I SIA, per quanto riguarda le soluzioni proprietarie, vengono sviluppati da

aziende specializzate che si occupano non solo dell'implementazione dei prodotti ma anche della distribuzione, dell'assistenza e della manutenzione degli stessi. Queste aziende, spesso anche tramite l'ausilio di una rete di rivenditori autorizzati, supportano il cliente sia in fase d'introduzione del nuovo sistema, che durante l'utilizzo, ne forniscono aggiornamenti e, a seconda dei casi, effettuano personalizzazioni su richieste del cliente.

Il mercato ERP oggi si è evoluto in assoluta oligarchia. Per poter offrire ai clienti soluzioni sempre avanzate rispetto allo stato dell'arte le aziende produttrici si trovano a sostenere costi elevatissimi, dovuti anche ad un'evoluzione tecnologica inarrestabile. Ciò ha collassato la piazza ad un consesso di oligarchi che determinano il mercato.

I grandi produttori dominano il mercato delle multinazionali e grandi imprese nazionali ma hanno minor penetrazione fra le PMI (Piccole e Medie Imprese), dove sono presenti maggiormente soluzioni non proprietarie.

Diversamente da ciò che avviene con le soluzioni proprietarie, le soluzioni libere si caratterizzano per il supporto fornito dalla comunità di sviluppo e per una maggiore libertà di scelta per l'utente finale. Con le soluzioni libere, infatti, non si è legati ad una casa produttrice di software bensì qualunque azienda, acquisite le competenze necessarie, può fornire assistenza e supportare il cliente, nonché sviluppare personalizzazioni per esso[4].

Il mercato attuale degli ERP si sta spostando verso le PMI, sia per la saturazione del mercato delle imprese di maggiori dimensioni sia per l'interesse che gli ERP ormai suscitano anche per le PMI. Quindi, la vera battaglia per il

futuro si combatte nel mercato PMI, territorio certo non vergine per le soluzioni ERP ma dai confini vastissimi e dal potenziale elevatissimo.

Nel paragrafo 1.4 si discutono le esigenze delle imprese che rientrano in questa categoria; di seguito, invece, vengono presentati alcuni fra gli ERP attualmente più diffusi sul mercato. Sono presentate soluzioni commerciali ed open source.

1.3.1 SAP: una soluzione commerciale

Tra i grandi produttori troviamo SAP². Con la sigla SAP ci si riferisce alla linea di prodotti software sviluppato dalla multinazionale tedesca SAP AG, uno dei principali leader mondiali nel settore degli ERP e, in generale, nelle soluzioni Enterprise.

Tramite un'ampia possibilità di parametrizzazione di ogni componente, al cliente è consentita una notevole modularità di funzionamento, in grado di rispondere a esigenze complesse come quelle delle multinazionali. Il tutto senza troppi vincoli con il particolare settore d'affari o la distribuzione geografica delle società, i siti produttivi e di distribuzione. Il prodotto soddisfa le esigenze e le normative legali e fiscali delle principali nazioni, occidentali ed orientali.

Una scelta, fin dall'inizio presente nei prodotti SAP, che si è rivelata vincente, riguarda l'astrazione dalla piattaforma sia da un punto di vista hardware che software. Esiste uno strato software (scritto in linguaggio C) che

² <http://www.sap.com>, <http://www.sap.com/italy> (ultima consultazione novembre 2010)

si interfaccia con i principali sistemi operativi e database consentendo al cliente una scelta della piattaforma a lui più congeniale ed una facile intercambiabilità della stessa.

SAP utilizza un'architettura three-tier basata su tre livelli: interfaccia utente, logica funzionale e persistenza dei dati.

Si tratta di un software di classe Enterprise costituito da un nucleo di componenti di base per l'integrazione e il funzionamento dei moduli applicativi. Il core, oltre ad un ambiente di personalizzazione posto a disposizione dei clienti, eroga funzioni essenziali e trasversali quali, tra le altre, login sul sistema, navigabilità, personalizzazione parametrica dei componenti, ampliamenti, gestione della base di dati. I moduli gestionali sono integrati e coprono le tematiche proprie di una grande impresa: dall'amministrazione finanziaria e alla tesoreria al controllo di gestione, dalla pianificazione alla gestione della logistica di merci e servizi, dalla vendita e distribuzione sino all'integrazione dei cicli di produzione.

Con mySap ERP il software presenta un'architettura di tipo SOA (Service Oriented Architecture). Con tale architettura le risorse sono disponibili agli altri partecipanti all'interno della rete in forma di WS (Web Service), accessibili in modo standardizzato. La SOA di SAP è chiamata ESA (Enterprise Service Architecture). L'idea di base è quella di rendere disponibili, per l'accesso dall'esterno, le funzioni interne della tecnologia e delle applicazioni SAP. Per poter creare un'architettura orientata ai servizi, combinando così i vantaggi delle applicazioni SAP con la flessibilità dei WS, è necessario disporre di un'adeguata piattaforma applicativa e tecnologica. In SAP tale piattaforma è

chiamata NetWeaver.

SAP permette una buona gestione aziendale, permettendo una forte diminuzione del tempo dedicato alle informazioni all'interno dell'azienda. Tuttavia presenta una forte rigidità che da un lato porta ad un'elevata sicurezza sulla correttezza delle informazioni, dall'altro può portare a dispendi di tempo da dedicare nell'analisi di tutto il percorso dell'informazione per risalire al problema. Questo non rende, inoltre, immediato l'apprendimento del funzionamento del sistema.

L'introduzione di un software di tipo ERP richiede una progettazione congiunta di tecnologia e organizzazione. Questi software sono prescrittivi per i processi, nel senso che impongono una trasformazione dei processi in funzione dello strumento informatico. La parte forse più impegnativa nell'introduzione e avviamento di SAP in azienda è la fase di studio e parametrizzazione. L'uso di software applicativi integrati comporta una notevole e ampia revisione critica dei processi organizzativi, ogni componente dovrà integrarsi con le altre.

1.3.2 Soluzioni open source

1.3.2.1 Openbravo

Openbravo³ è il leader in Italia, ed in tutto il mondo, degli ERP open source. È una soluzione free software, sviluppata e gestita da Openbravo SL.

³ <http://www.openbravo.com> (ultima consultazione novembre 2010)

È stato progettato con funzionalità di business di base ed è basato su un unico modello coerente in tutte le aree applicative che, insieme, fanno dell'applicazione un sistema completo di gestione aziendale. Le funzionalità che espone sono quelle tipiche di un ERP quali registro dei clienti ed impiegati, produzione, gestione acquisti e vendite, gestione progetti, gestione magazzino, gestione processi finanziari ecc.

Openbravo è stato sviluppato con il proposito di consentire ad esperti di business, con scarsa esperienza di sviluppo, di configurare facilmente l'applicazione. È uno strumento multi-piattaforma ed integrabile, tramite web service, con altri applicativi aziendali. Interamente sviluppato in Java, è configurabile in base alle necessità del cliente, velocizza i processi di business dandogli piena visibilità sempre ed ovunque, riduce i costi operativi ed è molto semplice da personalizzare rispetto ai suoi diretti concorrenti a pagamento.

È una piattaforma open source integrata ed orientata al Web. L'interfaccia utente si presenta amichevole e funzionale ed è stata progettata per essere utilizzata anche tramite touch screen.

Supporta hardware quali lettori di codici a barre, lettori di tessere magnetiche e registratori cassa.

1.3.2.2 Compiere

Un altro dei sistemi ERP open source più diffusi è Compiere⁴. Sviluppato in Java dalla Compiere, Inc., è pensata per le PMI.

4 <http://www.compiere.com> (ultima consultazione novembre 2010)

Compiere fornisce quattro edizioni di cui tre distribuite tramite licenza commerciale (Enterprise Edition, Standard Edition e Professional Edition) ed una tramite licenza open source (Community Edition) le quali differiscono non solo in termini di funzionalità offerte ma soprattutto in termini di supporto.

Compiere si caratterizza per la completezza delle funzionalità offerte. Consente infatti di affrontare in modo adeguato le tipiche esigenze aziendali. Integra le aree di contabilità, acquisti, produzione, vendite, gestione clienti e fornitori, distribuzione, logistica, gestione inventari e scorte di magazzino, risorse umane e qualsiasi altra funzione o nicchia aziendale. È in grado di gestire le attività di processo di una piccola e media impresa, fornisce funzionalità di gestione dei dati, strumenti di analisi delle informazioni e di supporto al processo decisionale, creazione ed esportazione di report.

È orientato al web ed eredita i vantaggi del linguaggio in cui è stato sviluppato, quindi è multi-piattaforma e non vincolato ad un solo DBMS. È stato sviluppato con un focus per processi piuttosto che per funzioni. Tale approccio fa sì che l'applicazione agisca in modo trasversale rispetto alle singole funzioni e che le automatizzi in modo integrato. Si basa su principi di interoperabilità interna ed esterna che permette un facile dialogo tra i diversi moduli ed i sistemi informativi esterni, permettendo una comunicazione tra i propri moduli applicativi e quelli di altri fornitori e/o costruiti ad hoc. È basato su un dizionario dati, il che consente un elevato grado di personalizzazione adattandosi alle diverse esigenze di ogni azienda. Offre, inoltre, funzioni in grado di monitorare il raggiungimento di obiettivi strategici ben definiti, tramite indicatori di vario genere, ed il tutto in tempo reale, supportando eventuali cambiamenti degli

obiettivi in corso d'opera.

1.4 I sistemi informativi per le PMI

Per PMI si intende una categoria di aziende le cui dimensioni rientrano entro certi limiti occupazionali e finanziari prefissati. Una media impresa è definita come un'impresa il cui organico sia inferiore a 250 persone e il cui fatturato non superi 50 milioni di euro ovvero il cui totale di bilancio annuale non sia superiore a 43 milioni di euro. Una piccola impresa, invece, è definita come un'impresa il cui organico sia inferiore a 50 persone e il cui fatturato ovvero il totale del bilancio annuale non superi 10 milioni di euro⁵.

Una PMI non ha minore complessità di una grande azienda. Ha una struttura interna orientata ai processi, spesso le stesse persone rivestono ruoli diversi, interfunzionali, senza una precisa formalizzazione della posizione assunta. D'altra parte ha esigenze molto più limitate ad un settore specifico, un volume di utenza più ristretto e quindi una minore complessità geografica. Le PMI, diversamente dalle grandi imprese, presentano una struttura meno rigida, ossia più flessibile, quindi sono maggiormente propense al cambiamento. Tuttavia esse dispongono di minori risorse che ne limitano gli investimenti, anche in termini di innovazione tecnologica.

⁵ Definizione adottata dalla Commissione europea con Raccomandazione 361 del 8 maggio 2003.

Dunque anche le PMI, specie negli ultimi anni, decidono di avvalersi sempre più di soluzioni informatiche per la gestione dei processi aziendali. Esse, solitamente, scelgono di avvalersi di soluzioni, seppur già esistenti, ampiamente personalizzabili e, specie quando manca all'interno dell'azienda una figura professionale dedicata, prediligono quei fornitori con i quali instaurare relazioni personalizzate. L'introduzione di un sistema informatico all'interno di un'impresa però comporta di fatto una dipendenza verso tale sistema.

Quindi evolvere il SIA in una PMI non solo può comportare una modifica delle attività e dei processi interni dell'azienda, ma può arrivare a cambiarli radicalmente. Per questo motivo le diversità strutturali ed organizzative che caratterizzano le varie aziende richiedono un diverso percorso di implementazione dei sistemi ERP.

In particolare si prospettano due possibili soluzioni:

- adeguamento dei processi aziendali interni sulla base delle funzionalità offerte dal sistema,
- adeguamento ovvero implementazione del sistema gestionale per rispondere alle esigenze dell'azienda.

In entrambi i casi, si determina un notevole impatto sull'organizzazione aziendale che necessita di essere gestito tramite un'accurata pianificazione del processo di cambiamento.

1.5 Evoluzione dei sistemi informativi: eventi interni ed esterni

Uno dei principali obiettivi di un'impresa è raggiungere un elevato livello di efficienza. Le motivazioni che inducono un'impresa ad evolvere il proprio sistema informativo[5] possono essere ricondotte a due categorie di eventi.

- Eventi esterni: aumento della pressione del mercato, modifica della normativa vigente, cambio dei rapporti o delle condizioni in essere con fornitori o clienti, ecc.
- Eventi interni: innovazione tecnologica, ottimizzazione e razionalizzazione dei processi produttivi, spirito di innovazione, ecc.

La prima spinta evolutiva è causata dalla migliore efficienza relativa dei concorrenti. Questa induce le aziende a cambiare per rimanere competitive, evitando il rischio di pagare il prezzo dell'inefficienza venendo estromesse dal mercato stesso. Infatti, qualora si presenti la possibilità di incrementare la propria efficienza, molto dipende dalla velocità con cui i concorrenti rincorrono l'evoluzione e quanto questo consenta loro di aumentare la pressione del mercato. Tra gli altri fattori esterni, si annovera ad esempio la modifica della normativa vigente, evento questo che può rendere obbligatorio per l'impresa l'adozione di nuovi processi o la modifica di quelli in essere. Causa di avvio del processo di modernizzazione può anche essere la variazione degli accordi in essere con fornitori o clienti. La scadenza di una licenza software, o il venire meno del supporto ad un software attualmente adottato dall'impresa, pone nelle condizioni di doverlo sostituire. Spesso non è possibile trovare un prodotto con

le medesime caratteristiche in grado di operare esattamente nello stesso modo, e ciò comporta la necessità di avviare un processo di evoluzione, andando ad influenzare tutte le procedure in cui il software era coinvolto.

Tra le cause interne di evoluzione di SIA si annovera invece l'innovazione tecnologica. Ogni qual volta le nuove tecnologie rendono disponibili, ed economicamente convenienti, soluzioni in grado di aumentare l'efficienza complessiva l'impresa è naturalmente spinta al cambiamento. L'evoluzione raggiunta a livello organizzativo, quindi, è spesso dipesa dalle nuove possibilità tecnologiche e dal calo dei costi relativi agli strumenti informatici, e di fatto, comporta un cambiamento ed una rivisitazione delle procedure interne aziendali. La velocità con cui l'azienda mette in atto il processo di evoluzione, ogni qual volta l'innovazione tecnologica lo renda possibile, dipende da fattori interni quali le risorse disponibili e lo spirito di innovazione del personale in grado di intraprendere ed influenzare le scelte inerenti le evoluzioni. Infatti, le aziende con personale ad elevato livello di esperienza in ambito IT si ritrovano ad avviare un processo di modernizzazione anche senza apparente necessità, migliorando di fatto la propria efficienza. Un'azienda che cerca di rimanere sempre al passo con i tempi, da un punto di vista tecnologico, avrà il vantaggio di non dover sopportare l'impatto di una radicale, o comunque troppo spinta, evoluzione del sistema; inoltre saranno più abituate al cambiamento e quindi in grado di gestirlo meglio.

Di fatto, però, le imprese mostrano una scarsa capacità di gestire l'evoluzione, tale incapacità è dovuta alla necessità di realizzare il cambiamento integrandolo con le ordinarie attività produttive senza causare interruzioni del

flusso produttivo.

Un altro dei maggiori problemi è la modifica dell'organizzazione aziendale ovvero l'aggiornamento dei processi interni in modo da adottare le procedure previste dal nuovo sistema.

Infine, si corre il rischio di adottare soluzioni troppo rigide, ovvero non in grado di agevolare successive evoluzioni del sistema che potrebbero in futuro rendersi necessarie.

Capitolo 2

Stato dell'arte

In questo capitolo sono analizzati alcuni framework fra quelli definiti in letteratura per la gestione del cambiamento, applicabili, quindi, al processo di evoluzione di SIA. L'obiettivo è di analizzare le metodologie per trarne spunto nella definizione di un nuovo framework adatto al contesto delle PMI.

2.1 Introduzione

Un framework consiste in una serie di passi, strategie e linee guida[6] che consentono di guidare le persone nella gestione del processo e coloro che in qualche modo ne sono coinvolte.

I framework presentati si caratterizzano per l'affrontare il problema

dell'evoluzione da punti di vista differenti. Alcuni suggeriscono una serie di attività da svolgere per l'implementazione del cambiamento concentrandosi sul sistema da un punto di vista tecnico, descrivendo modelli e strumenti per rappresentare e definire il sistema, nonché strumenti per trasformare il codice a basso livello in un meta-modello. Un altro framework analizza la questione in senso più ampio, studiando il problema tenendo conto dell'intero sistema azienda, considerando questa un insieme di sistemi e persone, nonché entità esterne, collegate tra loro. Un altro ancora analizza il problema della manutenzione e quindi del processo di aggiornamento del software. L'ultimo invece, si preoccupa di analizzare alcuni aspetti inerenti il cambiamento, in particolare quello della resistenza opposta dai potenziali utilizzatori, e suggerisce le strategie per arginare tale fenomeno.

L'analisi conclusiva è finalizzata a rivedere tali metodologie con particolare attenzione alle che sono le realtà delle PMI, valutando quindi, il grado di applicabilità dei framework descritti.

Nell'ultimo paragrafo si presentano brevemente altri lavori analizzati non ritenuti utili allo scopo e quindi non ulteriormente approfonditi.

2.2 Architecture-Driven Modernization

Il consorzio OMG[7] (Object Management Group) sta lavorando ad uno

standard chiamato ADM[8] (Architecture Driven Modernization) che sistematizza il processo di modernizzazione grazie all'ausilio di standard. Definisce ADM (Architecture-Driven Modernization) come il processo di analisi ed evoluzione di risorse software esistenti allo scopo di migliorare il software, ovvero effettuare modifiche ad esso.

ADM nasce con lo scopo di:

- re-ingegnerizzare le applicazioni esistenti;
- rendere il software flessibile ed adattabile;
- espandere ed influenzare gli standard MDA[9] (Model Driven Architecture);
- consolidare le pratiche di modernizzazione, per minimizzare il rischio durante il processo;
- incrementare di produttività nello sviluppo del software con uno sforzo minimo ed a basso costo.

Nonostante la varietà di scenari di un progetto di modernizzazione ADM classifica il processo in base a tre prospettive architettureali appartenenti a due domini[10]:

1. architettura tecnica,
2. architettura di applicazioni / dati,
3. architettura di business.

Come illustrato in Figura 4, la prime due rientrano nel dominio IT (Information Technology), l'ultima nel dominio di business; il dominio IT

comprende l'architettura da un punto di vista tecnico, i dati e le applicazioni, mentre il dominio di business comprende modelli, diagrammi dei processi di business e regole.

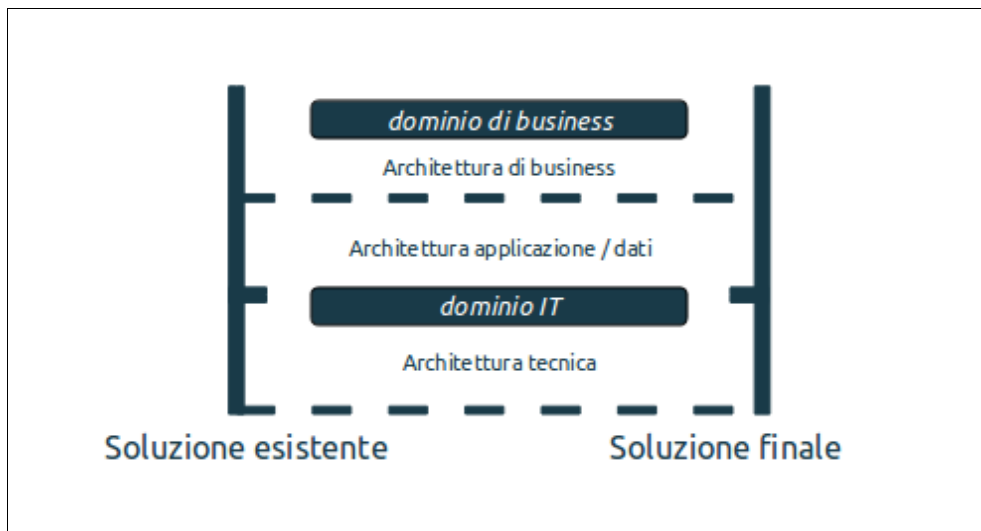


Figura 4: architetture nei domini di Business e IT

La trasformazione dell'architettura consiste in un insieme di progetti per la migrazione, anche incrementale, di porzioni della soluzione esistente verso la nuova. Questi progetti possono essere eseguiti al livello del dominio di business o al livello del dominio IT. Dunque, in base al livello di astrazione, si parla di:

1. modernizzazione guidata dall'architettura tecnica,
2. modernizzazione guidata dall'architettura di applicazioni / dati,
3. modernizzazione guidata dall'architettura di business,

come raffigurato in Figura 5.

La modernizzazione legata dall'architettura tecnica consiste, ad esempio, nella migrazione di un sistema software da una piattaforma ad un'altra o tra due

linguaggi⁶. Invece, la modernizzazione a livello di applicazioni e dati richiede cambiamenti anche al livello tecnico. La modernizzazione a livello di business, infine, è la più complicata e completa in quanto influenza i modelli a tutti i livelli di astrazione.

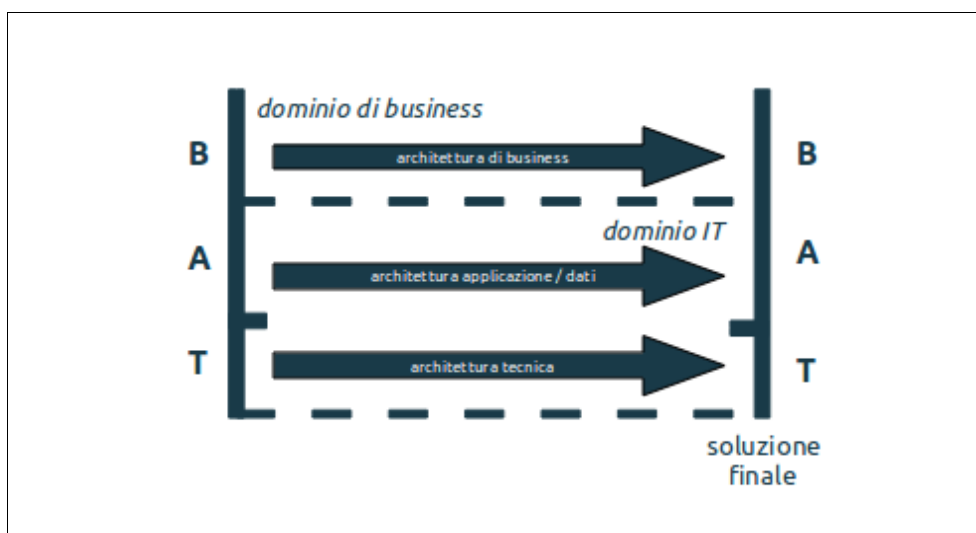


Figura 5: processi di modernizzazione

A prescindere dal livello architeturale in cui la modernizzazione impatta ci sono tre elementi da considerare nel processo di trasformazione che porta dalla soluzione corrente a quella nuova: la scoperta ed analisi della soluzione esistente, la definizione dell'architettura della soluzione finale e i passi per la migrazione (incrementale) dallo stato di partenza a quello finale.

Quindi il processo di ADM, rappresentato in Figura 6, inizia con l'applicazione di tecniche di reverse engineering, per raccogliere ed organizzare informazioni sulla soluzione attuale, ed una fase di forward engineering in cui si migra verso la soluzione definitiva. La modifica delle procedure avviene al livello

⁶ La modernizzazione a questo livello dovrebbe essere distinta da quella a livello di applicazioni e dati ma, spesso, modifiche a questo livello implicano modifiche sulle applicazioni.

di business[11].

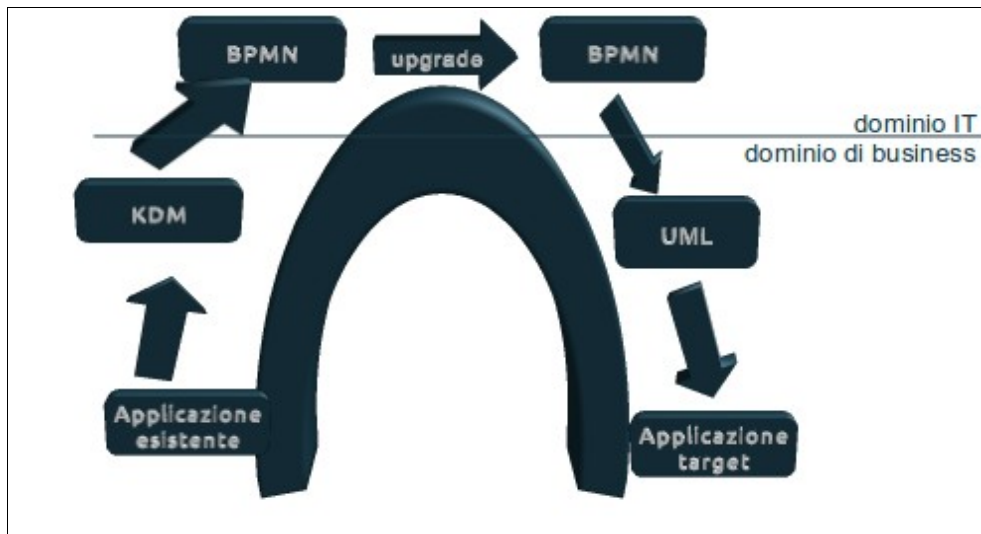


Figura 6: diagramma a ferro di cavallo del processo ADM

Il processo è supportato fornendo specifiche, formalismi e linee guida per l'analisi delle risorse software esistenti e la definizione dei meta-modelli.

KDM[12] (Knowledge Discovery Meta-Model) è un meta-modello che rappresenta informazioni circa le risorse software esistenti e l'ambiente operativo. È utilizzato per lo scambio di meta-dati sulle applicazioni, fra strumenti di modernizzazione di diversi produttori, e tra diversi linguaggi e piattaforme.

ASTM[13] (Abstract Syntax Tree Meta model), invece, fornisce una rappresentazione universale dei linguaggi di programmazione. Permette di modellare sistemi esistenti al livello procedurale poiché consente una corrispondenza diretta fra le istruzioni ed il modello del software di basso livello.

2.3 Model-Driven Software Modernization

MDSM[14] (Model-Driven Software Modernization) descrive un approccio per la modernizzazione di sistemi legacy, spesso utilizzato in progetti di modernizzazione esistenti ma non ancora sistematizzato. Lo scopo è quello di ridurre l'effort dovuto al cambiamento.

MDSM è definito come un processo per la modernizzazione, parzialmente automatizzata, di codice legacy basato sul modello e sulla trasformazione dei modelli. L'approccio enfatizza la riusabilità del processo e dei componenti dell'infrastruttura; predilige, ove possibile, un elevato livello di automazione; mira ad aumentare la qualità del codice; tende al mantenimento delle funzionalità esistenti.

Si basa sui concetti di:

- meta-modello, per la definizione e la gestione dei modelli;
- trasformazione del codice in meta-modelli e viceversa;
- trasformazione da un modello all'altro.

Utilizza per lo scopo strumenti per:

- rappresentare i modelli;
- la scoperta dei modelli;
- manipolare i modelli;
- generare codice.

MDSM assume che il vecchio sistema venga completamente dismesso ovvero sostituito con la nuova versione; inoltre si concentra in modo particolare sullo sviluppo degli strumenti per la migrazione di cui sopra, partendo dal presupposto che non esistono soluzioni utilizzabili senza bisogno di essere adattate alle proprie esigenze.

Di seguito i passi su cui si basa il processo di modernizzazione:

1. analisi dei requisiti;
2. sviluppo degli strumenti di migrazione, identificazione, acquisizione ed integrazione;
3. preparazione della piattaforma target;
4. pianificazione di un processo di migrazione automatizzato;
5. esecuzione della migrazione, con eventuali correzioni manuali;
6. verifica.

MDSM identifica, inoltre, due approcci che si differenziano, in particolare nel secondo passo.

- RMA (Reverse Model-Driven Architecture), si basa su più livelli di astrazione di meta-modelli per lo studio e la modifica del sistema.
- DTA (Direct Transformation Approach), che tende a semplificare il processo precedente.

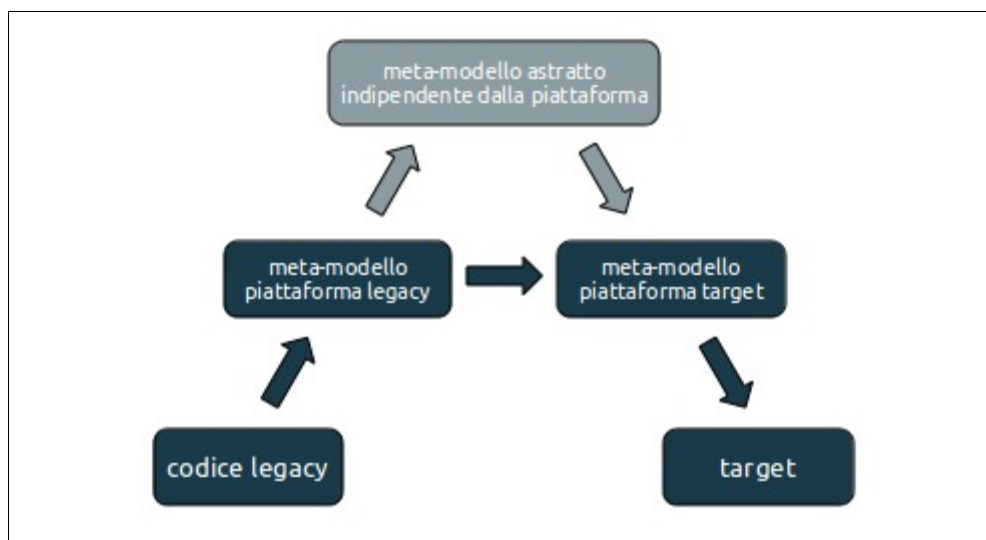


Figura 7: approcci MDSM: RMA e DTA

Questi approcci presentano attività simili ed un elevato numero di requisiti in comune. Come mostrato nella Figura 7 le attività in comune sono la scoperta del meta-modello e la generazione del codice per la piattaforma target. Il passaggio da un modello all'altro invece è differente. Infatti, il primo prevede un'ulteriore astrazione del modello, mentre il secondo effettua una trasformazione diretta da un modello all'altro.

2.4 Un framework basato sulla metodologia MDE di Alexandre Cláudio de Almeida et al.

Il framework descritto in [15] considera i seguenti aspetti di un SIA:

1. i dati processati per generare informazioni di business;

2. le funzionalità che trasformano i dati in informazioni;
3. le regole di business che stabiliscono le manipolazioni possibili sui dati.

L'idea di base è quella di incentrarsi sul modello concettuale di questi tre aspetti e generare in automatico il codice del software per ciascuno di essi. Le regole di business sono considerate un aspetto a sé e non, come spesso accade, un insieme di regole radicate nelle specifiche di dati e funzionalità.

Il framework si basa sulla metodologia MDE[16][17] (model-driven engineering), una metodologia di sviluppo del software che si focalizza sulla creazione di modelli relativi a particolari concetti del dominio di interesse, piuttosto che sugli aspetti algoritmici. Il framework, dunque, presenta un approccio per lo sviluppo e la manutenzione dei SIA basato sui seguenti modelli concettuali, in base ai tre aspetti di cui sopra:

1. schema dei dati,
2. modello delle funzionalità,
3. modello delle regole di business.

L'architettura del framework, come mostrato in Figura 8, è suddivisa nelle seguenti componenti:

- Metadata,
- Interface,
- Service,
- Business rule,

- Persistence.

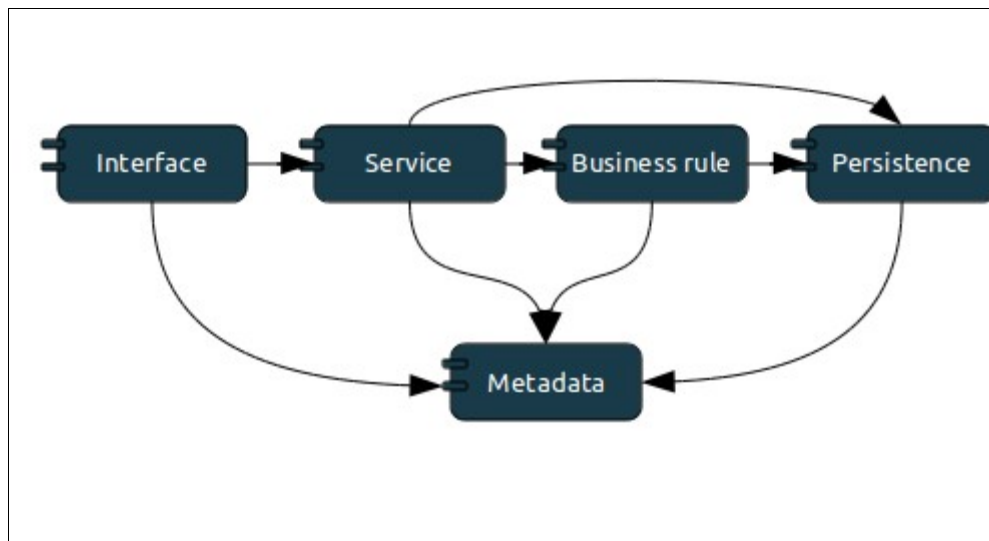


Figura 8: architettura delle componenti del framework per modellare, costruire e mantenere SIA

Il componente Metadata rappresenta il cuore del framework in quanto consente la gestione del modello concettuale del SIA. Il modello definisce i meta-dati utilizzati poi dagli altri componenti del framework, infatti è l'unico a non fruire dei servizi forniti dagli altri componenti.

Il componente Interface usa il modello concettuale per generare automaticamente l'interfaccia utente per le applicazioni.

Il componente Service fornisce strumenti e servizi per salvare e manipolare le informazioni provenienti dall'interfaccia verso le altre componenti del framework. Ad esempio, riceve i dati dal componente Interface per inviarli al componente Persistence.

Il componente Business Rule si occupa di gestire un archivio centralizzato, tramite il componente Persistence, di tutte le regole di business che in seguito sono mappate sullo schema dei dati.

Infine, il Persistence consente di mappare il modello concettuale del sistema nel modello operativo dei dati rispettando al contempo le regole di business.

Il framework consente, dunque, di gestire il processo di evoluzione di un SIA attuando la modernizzazione sul modello. L'esportazione delle modifiche sul sistema è effettuata poi tramite strumenti di trasformazione automatici.

2.5 Un framework basato sulla disciplina ESE di Robert S. Swarz e Joseph K. DeRosa

La ESE[18] (Enterprise Systems Engineering) è una disciplina emergente dell'ingegneria che studia l'integrazione ed i principi di un sistema complesso, composto da molti sottosistemi collegati tra loro. Il framework in analisi[19] definisce un'impresa non come una singola entità o semplicemente un sistema composto da sottosistemi, bensì come un insieme di sistemi le cui capacità operative sono inestricabilmente intrecciate con le considerazioni di persone, processi e tecnologie ed i cui confini sono spesso imprecisi. Identifica inoltre delle proprietà aggiuntive quali comportamenti emergenti⁷, non-determinismo e dipendenza dal contesto. La ESE, infatti, considera l'intero contesto in cui l'impresa opera, incluse le interazioni con le persone, la struttura di governo, la

⁷ I comportamenti emergenti si verificano in un insieme di sottosistemi i cui comportamenti individuali possono essere previsti in base alle caratteristiche individuali che, operando insieme, danno vita a comportamenti non prevedibili basandosi sulle caratteristiche dei singoli.

manutenzione dei sistemi ed il supporto. Il numero di interazioni cresce esponenzialmente con l'aumentare del numero di sottosistemi, andando ad incrementare la complessità dell'impresa ed i comportamenti non prevedibili. Lo studio dell'azienda è effettuato analizzando l'intero sistema in tutta la sua complessità, tenendo conto delle influenze esterne e dei comportamenti adattativi dei sottosistemi causati dalle relazioni esistenti fra essi.

Il framework delinea i seguenti cinque processi che hanno lo scopo di gestire e controllare l'evoluzione dell'impresa.

1. Technology Planning
2. Capability-Based Engineering Analysis
3. Enterprise Architecture
4. Strategic Technical Planning
5. Enterprise Analysis and Assessment

Questi processi, dunque, delineano un framework per applicare l'ingegneria dei sistemi al contesto delle imprese. Per ogni processo sono forniti una serie di attività con lo scopo di guidare il processo di sviluppo dei componenti del sistema.

Il primo processo consiste nell'analizzare i trend attuali, sia in ambito commerciale che fra la comunità di ricerca, in modo da identificare la tecnologia che soddisfa maggiormente i bisogni del sistema. Nello scegliere la tecnologia bisogna tener conto dell'applicabilità, della maturità della stessa, del supporto disponibile, dell'affidabilità, della manutenibilità e dell'interoperabilità, sempre facendo riferimento alle necessità che si devono soddisfare. Bisogna poi tener

conto della transizione tecnologica che essa comporta, ossia dell'impatto dovuto all'integrazione della nuova tecnologia all'interno del resto del sistema.

Il secondo processo considera l'impresa come un insieme di risorse collegate, piuttosto che come un insieme di sistemi o programmi. Quindi tiene conto dell'impresa da un punto di vista globale, per giunta anche degli interessi di tutte le entità che in qualche modo hanno a che fare con l'impresa. Evolvere il sistema informativo di un'azienda può comportare, infatti, cambiamenti dei rapporti attualmente in essere con le entità esterne all'azienda con conseguente variazione del suo modello di business. Bisogna allora gestire tale cambiamento, ma bisogna anche analizzare i costi e l'impatto che le strategie evolutive hanno sull'azienda.

Il terzo processo consiste nel descrivere l'architettura dell'impresa illustrandone i componenti, di cui i ruoli, e le relazioni fra essi. In questa fase si descrivono sia l'architettura attuale che quella finale iniziando a pianificare i processi da attuare e a definire le tecnologie da utilizzare per la transizione.

Il quarto processo consiste nel pianificare le strategie per mettere in atto le evoluzioni tecnologiche. In questa fase si sviluppa una sequenza di passi per implementare l'evoluzione in modo incrementale. La pianificazione deve prevedere possibili modifiche in corso d'opera in base al feedback ricevuto già dalle prime fasi.

Infine, l'ultimo processo si occupa di verificare se la strategia e l'implementazione procedono come previsto. Consente, quindi, di misurare lo stato di avanzamento verso l'obiettivo analizzando il feedback ricevuto dalle

varie fasi.

2.6 *Un modello per la manutenzione di un ERP di Celeste See Pui Ng. et al.*

Il modello[20] descritto in questo paragrafo ha lo scopo di definire delle linee guida per la manutenzione e il processo di evoluzione di sistemi ERP. Esso mira a definire un approccio concretamente applicabile nella realtà.

Il modello propone una serie di attività da espletare a tal fine suddividendole in tre stage.

1. Maintenance Preparation Stage
2. Maintenance Procedure Stage
3. Software Upgrade Stage

I primi due si occupano di definire ed organizzare la manutenzione identificando i responsabili, chi se ne dovrà occupare, gli strumenti ed i software di supporto, ed impostandone le procedure. L'ultimo, invece, definisce le attività necessarie per l'aggiornamento di versione del software.

Le attività previste per questo processo sono:

1. definizione di un progetto per l'aggiornamento;
2. ricerca delle possibili opzioni per effettuare l'aggiornamento;

3. analisi e valutazione dell'aggiornamento;
4. valutazione delle modifiche apportate alla versione corrente;
5. valutazione delle nuove funzionalità disponibili sulla versione aggiornata;;
6. analisi dell'impatto;
7. installazione della versione aggiornata su una piattaforma di test / sviluppo;
8. applicazione delle modifiche, effettuate alla versione corrente, al sistema aggiornato;
9. test del sistema aggiornato;
10. applicazione dell'aggiornamento al sistema corrente.

Le linee guida proposte consentono di guidare un'azienda durante il processo di aggiornamento, e quindi passaggio alla nuova versione, del prodotto ERP in uso tenendo conto delle modifiche apportate alla versione corrente del prodotto.

2.7 Un framework per la gestione del cambiamento di Adel M. Aladwani

Nei paragrafi precedenti sono stati descritti alcuni framework per la

gestione del processo di migrazione, ma nulla è stato detto riguardo le strategie di introduzione del nuovo sistema all'interno di un'organizzazione. A tale proposito va innanzitutto evidenziato come, nonostante i benefici introdotti, l'adozione di molti sistemi ERP fallisce a causa della resistenza posta dai potenziali utilizzatori di fronte il nuovo sistema. I dirigenti dell'organizzazione dovrebbero quindi attuare una serie di interventi pro-attivi in modo da gestire tale fenomeno. In [21] viene studiato proprio il fenomeno della resistenza da parte dei dipendenti all'introduzione di un nuovo sistema ERP all'interno di un'organizzazione aziendale analizzandone le similitudini con le strategie di marketing.

Le strategie atte all'implementazione di un nuovo sistema possono essere suddivise in organizzative, tecniche e rivolte alle persone. Le strategie organizzative riguardano lo sviluppo e la distribuzione, le tecniche per la gestione del cambiamento, la gestione dei progetti, la struttura organizzativa e delle risorse, nonché gli stili e le ideologie manageriali, la comunicazione e la coordinazione. Le strategie tecniche, invece, riguardano gli aspetti tecnici quali l'installazione del software, la complessità dello stesso, la presenza di competenze tecniche interne, i tempi e i costi per l'implementazione. Infine, le strategie rivolte alle persone, studiano le attitudini attuali dei dipendenti, il coinvolgimento ed il training.

Per quanto riguarda le strategie di marketing ci sono vari filoni di ricerca fra cui quelli che si concentrano sul marketing strategico e quelli che studiano il comportamento del consumatore. Il primo suggerisce le attività che deve svolgere un'impresa per garantirsi la sopravvivenza, quali l'identificazione degli

obiettivi e la definizione delle strategie, in base a quelli che sono i bisogni del consumatore, per raggiungerli, la messa in opera di tali strategie e la verifica. L'altro filone di ricerca pone il focus sul comportamento del consumatore e basa il suo studio sui fattori interni ed esterni che ne influenzano le scelte in fase di acquisto. Per individuare tali atteggiamenti gli studiosi analizzano le componenti cognitiva, emotiva e conativa⁸.

Le strategie descritte presentano una serie di similitudini.

- Entrambe rappresentano un processo di scambio fra due parti.
- Gli elementi coinvolti sono essenzialmente gli stessi. Nel marketing questi sono l'acquirente, il venditore ed il prodotto; dal punto di vista del ERP, essi sono i potenziali utilizzatori, lo sviluppatore del ERP e il sistema ERP stesso.
- Entrambe le prospettive risentono del problema della resistenza al cambiamento.

È chiaro, dunque, che si possono sfruttare le strategie di marketing per gestire la resistenza degli utenti all'introduzione di sistemi ERP. Le strategie proposte prendono forma in un framework concettuale, orientato ai processi, che ha l'obiettivo di assistere il top management di un'impresa nel complesso problema organizzativo della resistenza dei dipendenti all'adozione di un nuovo sistema ERP. Le fasi in cui si sviluppa, riportate in Figura 9, sono le seguenti:

1. formulazione della conoscenza;
2. implementazione della strategia;

⁸ Che induce il desiderio.

3. controllo e valutazione delle strategie.

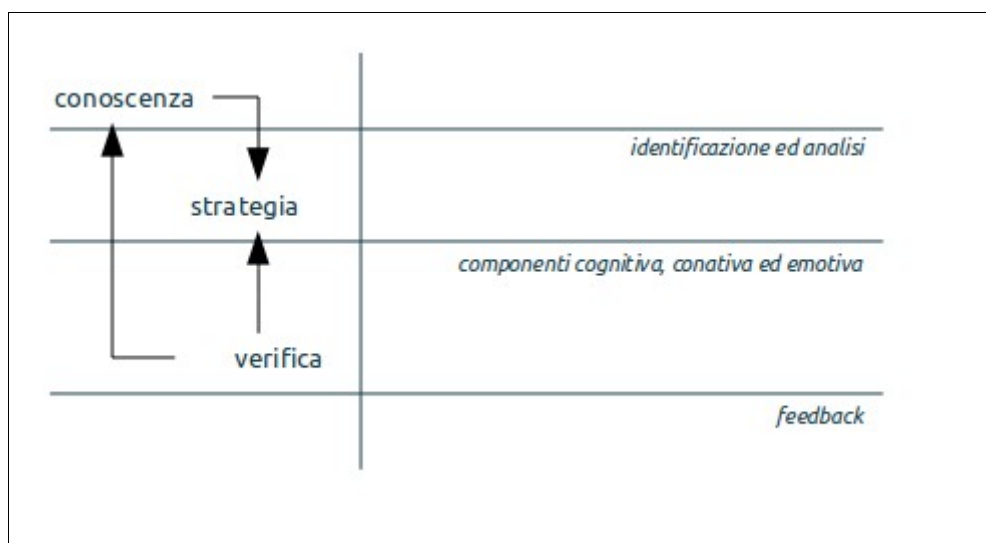


Figura 9: framework per la gestione del cambiamento

La prima fase consiste nell'identificazione e nell'analisi degli atteggiamenti delle persone ovvero dei gruppi più influenti all'interno dell'organizzazione. Una volta individuati i gruppi o le persone che possono eventualmente opporre resistenza, se ne analizzano i bisogni, i principi, le convinzioni e gli interessi.

Nella seconda fase il management può sfruttare la conoscenza acquisita, relativa ai potenziali utenti, per mettere in atto le strategie che consentono di sconfiggere la resistenza verso il nuovo sistema e convincere quante più persone è possibile ad utilizzarlo. In questa fase, relativamente alle strategie di marketing che si concentrano sul comportamento delle persone, lo scopo è quello di colpire le componenti cognitive, emotive e conative. Per avere effetti sulla componente cognitiva la strategia migliore è improntata sulla comunicazione, ovvero informare i potenziali utilizzatori dei benefici introdotti dal nuovo sistema, descriverne i benefici che ne trarrà l'azienda, illustrarne il funzionamento, proprio come, per convincere una persona ad acquistare un

prodotto, se ne descrivono il funzionamento ed i vantaggi. La componente emotiva, invece, può essere colpita con la strategia della minimizzazione dei costi, ovvero della minimizzazione dell'impatto. Il modo migliore di convincere una persona ad utilizzare un prodotto è convincerlo che il costo dovuto all'adozione del nuovo sistema è minimo, proprio come nelle strategie di marketing si tende a minimizzare i costi. Infine, come con i prodotti, le persone si fanno coinvolgere dagli altri, anche quando si deve adottare un nuovo sistema gli utenti si lasciano coinvolgere dalle cosiddette figure leader presenti all'interno dell'organizzazione.

L'ultima fase del framework consiste nel monitorare e valutare le strategie messe in atto, così da fornire al top management un feedback periodico, sistematico e accurato, sia riguardo l'introduzione del nuovo sistema sia riguardo al costo dovuto alla gestione del cambiamento. Se le informazioni raccolte dovessero essere negative allora bisogna spendere altre risorse per analizzare cosa è andato storto, ed eventualmente ripetere alcune delle fasi precedenti.

2.8 Le metodologie a confronto

Per concludere si confrontano le quattro metodologie sopra descritte rapportandole alla realtà in cui operano le PMI, tenendo conto della struttura

organizzativa e delle esigenze delle stesse.

ADM è un progetto piuttosto articolato e complesso che fornisce standard per la rappresentazione dei meta-modelli ma presuppone l'esistenza di strumenti di trasformazione semi-automatici che, nella maggior parte di casi, non esistono o non sono applicabili. Esso, però, non è adatto alla realtà delle PMI italiane in quanto non tiene conto delle problematiche che interessano tale tipo di imprese, ovvero, anche se ritiene possibile una migrazione incrementale non si focalizza su questo aspetto, fondamentale nelle realtà concrete dove i processi di trasformazione non devono comportare tempi di inattività dei sistemi. Infine, il processo non pone particolare attenzione alla scelta del sistema target, infatti potrebbe non esistere un sistema in grado di mappare direttamente le necessità emerse al livello di business, o si potrebbe non essere interessati a creare un ERP da zero.

In MDSM emerge che spesso non sono necessari troppi livelli di astrazione del sistema ma risulta più agevole un approccio diretto di trasformazione del modello. C'è da dire, ancora, che nella pratica non sempre esistono strumenti di trasformazione appropriati, quindi una trasformazione in meno è sicuramente meno work-intensive, anche in funzione di non dover sviluppare gli strumenti per eseguire la modernizzazione. In realtà MDSM anche se individua gli strumenti non ne fornisce. Identifica l'innovazione solo da un punto di vista tecnologico, senza considerare eventuali cambiamenti dei requisiti resi possibili dall'introduzione di un nuovo sistema. Risolve il problema dovuto alla non esistenza di un sistema atto a mappare le procedure aziendali solo con la realizzazione del sistema in base alle esigenze, senza prendere in

considerazione la soluzione inversa ovvero il cambiamento delle procedure. Infine, secondo MDSM il vecchio sistema viene dismesso del tutto, cosa non sempre vera nella pratica.

Il framework basato sulla metodologia MDE propone una classificazione delle componenti di un SIA che rende più agevole l'astrazione del sistema in un modello. Esso, una volta rappresentato il modello del sistema, secondo le componenti illustrate dal framework, ne consente lo sviluppo e le successive evoluzioni tramite l'ausilio di strumenti automatici per la generazione del codice. Il modello, però, si focalizza su un'evoluzione di livello troppo basso senza tenere conto degli aspetti legati alle procedure di business. Inoltre, l'unica evoluzione presentata è sulla base di dati, e non sono forniti esempi pratici sull'applicazione di strumenti automatici alle funzioni applicative.

Nel framework basato sulla disciplina ESE l'impresa è considerata come un insieme di sottosistemi. Si tiene conto, cioè, delle relazioni che intercorrono fra essi, delle persone e delle interfacce che utilizzano i sistemi, ma anche del mondo esterno e dei comportamenti emergenti dalle relazioni fra le entità sia interne che esterne. I processi derivanti da questo modello forniscono un supporto notevole poiché tengono conto di ogni aspetto legato all'evoluzione. Purtroppo però, spesso nelle imprese non è presente un'organizzazione aziendale adeguata, che consenta cioè da una parte di gestire ogni attività prevista dalla metodologia con un'appropriata ripartizione delle responsabilità. e dall'altra capace di effettuare una pianificazione così come previsto dai primi tre processi del framework.

Il modello di manutenzione presentato, invece, definisce un modello per la

manutenzione di un prodotto ERP. Di questo si analizza solo la parte che si occupa del processo di aggiornamento. I passi indicati sono abbastanza esaustivi ed espliciti e consentono di affrontare la questione minimizzando i rischi. La metodologia, però, si riferisce solo all'aggiornamento di versione dello stesso prodotto software senza prendere in considerazione la possibilità di adottarne un altro. Si limita quindi a cercare gli aggiornamenti disponibili per il prodotto in uso senza considerare eventuali altre necessità degli utenti. D'altro canto il processo di aggiornamento può essere inteso sì come evoluzione ma sicuramente in senso più ristretto rispetto al complesso processo di evoluzione dato che ne considera solo alcuni aspetti.

L'ultima metodologia non affronta la questione dal punto di vista tecnologico, bensì considera il fenomeno della resistenza degli utenti al cambiamento. Si tratta da un fenomeno che spesso viene sottovalutato dalle organizzazioni, tuttavia anche il miglior sistema non produrrà vantaggi finché non sarà accettato positivamente dai membri dell'azienda. Le strategie proposte sono ben dettagliate e utili per le imprese, ma la metodologia si concentra solo sull'aspetto descritto senza fornire alcun aiuto sulle scelte tecnologiche pur sempre di rilevanza fondamentale nel processo di evoluzione di un sistema informativo aziendale.

Di seguito vengono individuate una serie di aspetti ritenuti rilevanti ai fini dell'analisi svolta nel seguente lavoro e successivamente si valuta quanto le metodologie analizzate tengono conto di tali proprietà.

I. Usabilità

II. Livello di dettaglio

III. Supporto tecnico

IV. Contestualità

V. Incrementalità

L'usabilità evidenzia quanto il modello e le attività proposte siano concretamente applicabili. Il modello ha un basso livello di usabilità se propone soluzioni non facilmente realizzabili, troppo complesse ovvero astratte, oppure se effettua assunzioni utopiche.

Il livello di dettaglio misura la completezza della soluzione proposta, in termini sia di fasi che di attività da espletare per ogni fase.

Si valuta poi il supporto tecnico, ossia gli strumenti sia formali che pratici forniti o indicati dal framework utilizzabili come supporto alle varie attività.

La contestualità indica quanto il framework tiene conto dell'ambiente esterno, ossia di quelle entità, sia interne che esterne all'impresa, che influenzano e/o sono influenzate dal processo di evoluzione, ma anche delle soluzioni attualmente offerte dal mercato in termini di prodotti disponibili adottabili come sistema target.

L'ultima proprietà, l'incrementalità, verifica quanto il modello applica un approccio di tipo incrementale per eseguire l'evoluzione.

Le valutazioni per i framework descritti sono riassunte nella Tabella 1 dove per ogni aspetto è stata effettuata una valutazione in base al grado di bontà: alto, medio, basso.

	Usabilità	Livello di dettaglio	Supporto tecnico	Contestualità	Incrementalità
ADM	bassa	alto	alto	bassa	bassa
MDSM	alta	basso	basso	bassa	bassa
Alexandre Cláudio de Almeida et al.	media	alto	medio	bassa	bassa
Robert S. Swarz et Joseph K. DeRosa	alta	medio	basso	alta	media
Celeste See Pui Ng. et al.	alta	medio	basso	bassa	bassa
Adel M. Aladwani	alta	medio	basso	media	media

Tabella 1: confronto fra i modelli analizzati

In base a tali valutazioni si misura quindi il livello di adeguatezza del framework per la gestione dell'evoluzione di un SIA di una PMI. Da questo punto di vista tutte le metodologie studiate sono carenti, alcune per quanto riguarda degli aspetti altre per altri. In particolare i primi due framework peccano uno per la praticità, l'altro per il livello di dettaglio. Il terzo propone un buon modello ma non presenta un'alta usabilità, né considera tutti gli aspetti inerenti l'evoluzione. Il framework basato sui processi di ESE e quello per la gestione del cambiamento, seppur ritenuti più adatti, non considerano tutti gli aspetti importanti per gestire al meglio il processo di modernizzazione. Per quanto riguarda il modello di manutenzione esso prevede alcuni task sicuramente applicabili al il processo di evoluzione ma ancor più degli altri non ne considera tutti gli aspetti.

Altra valutazione da fare è che le metodologie esaminate non scendono

nel dettaglio del sistema target da adottare, ovvero partono dal presupposto che sia possibile trovare un sistema target esattamente rispondente alle esigenze emerse dal modello di business, senza prevedere una fase di adeguamento. Tale attività è prevista solo nel framework basato sulla disciplina ESE.

Nel prossimo capitolo si viene allo scopo del presente lavoro, descrivendo un framework che cerca di tenere conto di tutti gli aspetti emersi dall'analisi di cui sopra.

2.9 Altre metodologie

Un accenno meritano, per completezza della trattazione, alcune metodologie incontrate durante l'attività di ricerca, tuttavia ritenute poco rilevanti ai fini del presente lavoro, quindi, non ulteriormente approfondite.

La prima presenta un modello di evoluzione di un SIA basata su un'architettura dinamica, mentre la seconda descrive un approccio che si focalizza in modo particolare sulla qualità del sistema.

2.9.1 Evoluzione dinamica

Il lavoro in analisi[22] affronta il problema dell'evoluzione ad un livello più

alto di astrazione grazie ad un processo di sviluppo basato sull'architettura. Lo scopo è di gestire l'evoluzione in quei casi in cui il sistema non può essere bloccato ed i cambiamenti devono essere eseguiti sul sistema in esecuzione sia per quanto riguarda evoluzioni pianificate che non.

La metodologia si basa su ArchWare⁹ ADL (Architecture Description Language), un linguaggio di descrizione formale dell'architettura. L'architettura del software è considerata come un insieme di nodi di tipo diverso connessi tra loro. I nodi rappresentano le componenti del sistema ed i connettori le relazioni fra di essi. Il sistema deve rispettare determinati vincoli e/o requisiti in modo da essere compatibile con ArchWare.

L'evoluzione viene gestita operando direttamente sul codice del sistema ma le modifiche vengono prima studiate ed analizzate sul linguaggio formale e i cambiamenti che ne conseguono vengono verificati prima di essere applicati.

Il limite di questo approccio sta nel fatto che esso si rivolge a quei sistemi che sono stati sviluppati fin dal principio utilizzando ArchWare ADL ed è difficile, spesso impossibile, adattare un sistema esistente per descriverlo in tale linguaggio.

2.9.2 Un approccio verso la qualità

Lo scopo principale dell'approccio descritto in [23] è quello di garantire un elevato livello di qualità dei SIA. Le aree che influiscono sulla qualità del

⁹ <http://www.arch-ware.org> (ultima consultazione novembre 2010)

sistema sono classificabili in aree di business ed aree tecniche. Le prime definiscono le strategie di business ed il processo di sviluppo, le seconde l'architettura e le funzionalità del sistema. Queste di solito sono isolate ma tutte influenzano la qualità globale del sistema finale.

Lo scopo del lavoro in analisi è individuare, quindi, le dipendenze tra le suddette aree così da gestire le conseguenze di ogni singola decisione.

Si tratta però di un lavoro ancora in fase di sviluppo che, nonostante preveda di affrontare la questione dell'evoluzione, allo stato attuale si concentra solo sul processo di sviluppo.

Capitolo 3

Un framework per la modernizzazione dei SIA di PMI italiane

In questo capitolo si descrive il framework definito per l'evoluzione dei SIA di PMI. Si trae spunto dalle metodologie analizzate nel capitolo precedente e, al contempo, si pone particolare attenzione a quegli aspetti da esse scarsamente considerati.

3.1 Obiettivi

L'obiettivo principale è quello di definire un framework usabile, pratico, che sia concretamente applicabile, con un buon livello di dettaglio per ogni fase e per le attività da espletare ad ogni fase. Si indicano gli strumenti da adottare per portare a termine le varie attività, le documentazioni da produrre o i formalismi

che si possono utilizzare. La metodologia inoltre tiene conto degli aspetti sia interni che esterni che in qualche modo influenzano il processo. Utilizza un approccio incrementale in modo da consentire una modernizzazione che non comporti periodi di inattività troppo lunghi per l'impresa.

Altro obiettivo del framework è guidare l'utente nella scelta di un sistema da adottare fornendo i passi per arrivare ad una scelta aderente alle proprie esigenze. Infatti, relativamente alla realtà di interesse si presuppone che le imprese, nella maggior parte dei casi, abbiano a disposizione un budget limitato da dedicare al processo di evoluzione ovvero non abbiano la necessità di realizzare, o farsi realizzare, un SIA ad-hoc esistendone di adatti, o adattabili in commercio.

La metodologia è stata formulata cercando di coniugare tali necessità, proprie delle PMI.

3.2 Descrizione del framework

Di seguito sono elencate le fasi previste dal framework schematizzate in Figura 10.

1. Analisi interna dei bisogni: analisi dell'ambiente interno dell'impresa e dei bisogni di essa.
2. Definizione del target: analisi delle soluzioni offerte dal mercato e

definizione del sistema più adatto alle esigenze dell'azienda.

3. Progettazione delle modifiche: definizione delle modifiche da apportare ai moduli del sistema per adeguarli alle procedure aziendali e/o delle modifiche da apportare alle procedure aziendali per conformarle al sistema.
4. Integrazione del sistema target: sviluppo delle modifiche al sistema target.
5. Sostituzione del sistema: dismissione del vecchio sistema e messa in esercizio del sistema target.
6. Verifica della modernizzazione: valutazioni finali e dismissione del vecchio sistema.

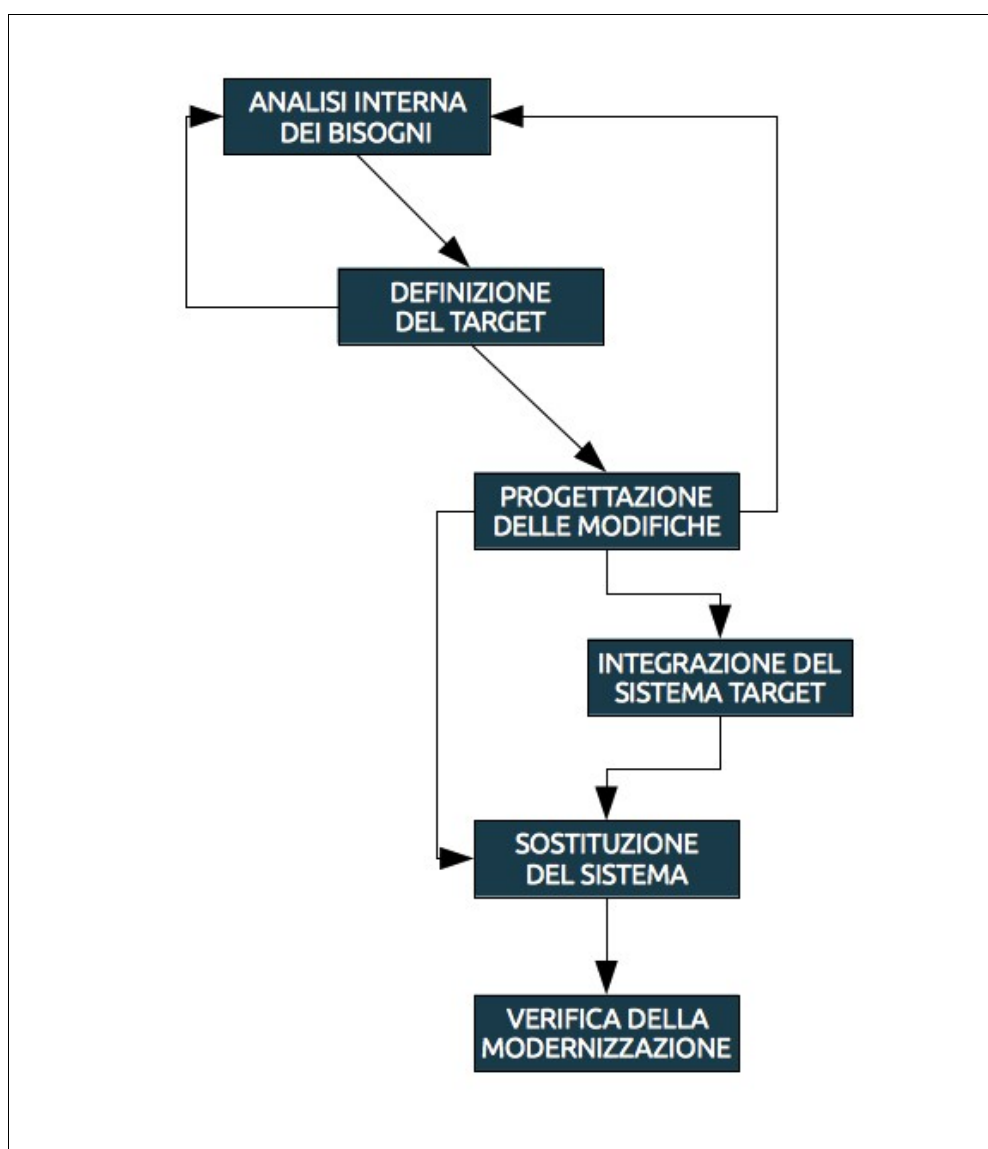


Figura 10: fasi del framework

Le prime tre fasi, di cui si riporta lo schema dettagliato in Figura 11, si occupano dell'analisi e della progettazione del cambiamento, mentre le successive, in Figura 13, riguardano l'implementazione dello stesso.

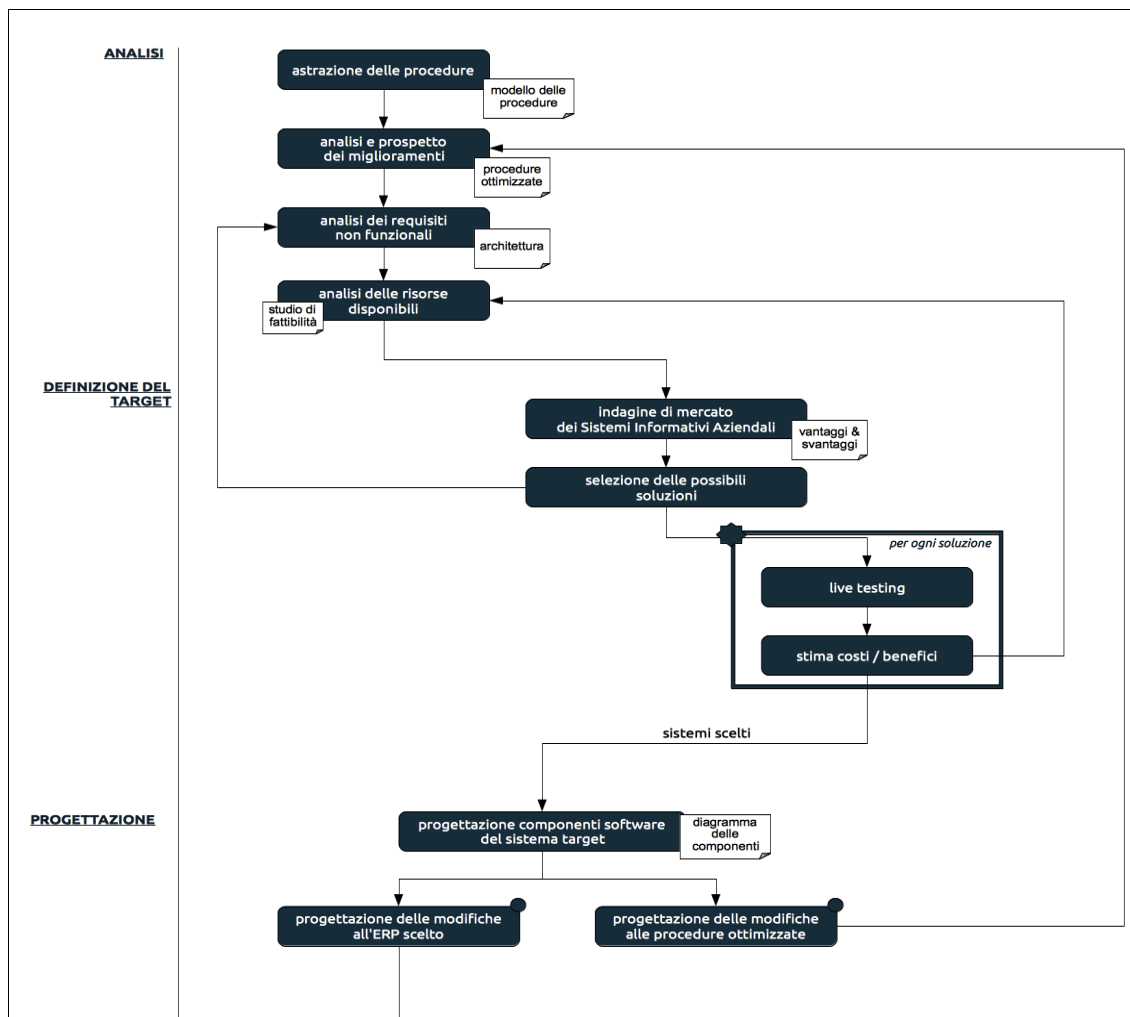


Figura 11: fasi di analisi e progettazione del cambiamento (vedi Figura 12)

Le attività di analisi consentono di raccogliere informazioni inerenti lo stato dell'arte dell'azienda. Per le attività da espletare si fa riferimento agli strumenti forniti dal framework ADM[8][10] (Architecture-Driven Modernization) studiato nel Paragrafo 2.2. Invece, come suggerito dall'approccio MDSM[14] (Model-Driven Software Modernization) illustrato nel Paragrafo 2.3, l'analisi volta all'ottimizzazione delle procedure si effettua su un modello indipendente dalla piattaforma.

La prima fase si articola nelle seguenti attività:

- 1.1. astrazione delle procedure;

- 1.2. analisi e prospetto dei miglioramenti;
- 1.3. analisi dei requisiti non funzionali;
- 1.4. analisi delle risorse disponibili.

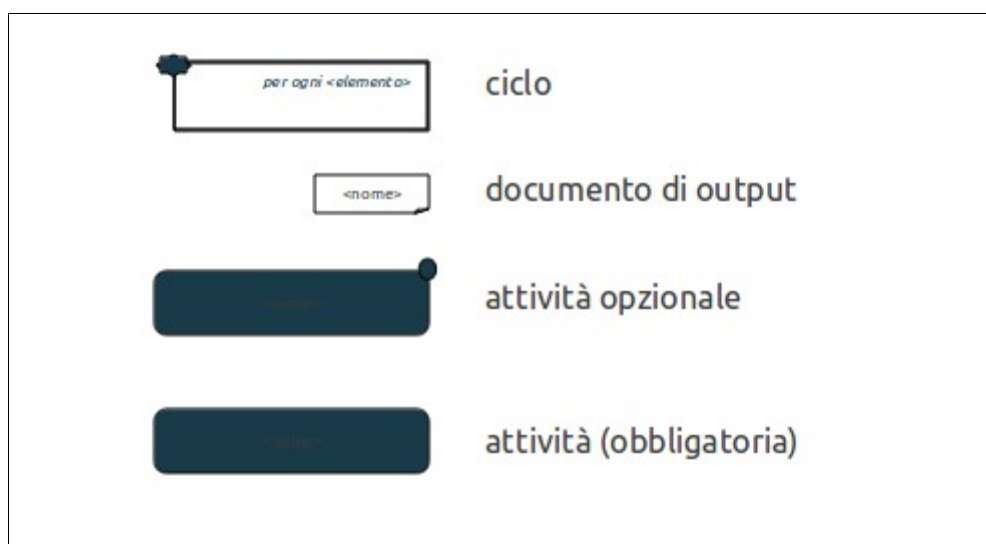


Figura 12: legenda schema framework

La prima attività si occupa di descrivere le procedure attualmente in essere mediante un modello semplificato e indipendente dalle tecnologie utilizzate. Si ottiene così un modello delle procedure con gli attori, i ruoli, le attività e le regole che guidano il flusso delle informazioni all'interno del processo.

Durante l'attività di analisi e prospetto dei miglioramenti vengono analizzate le procedure modellate e definiti i miglioramenti da effettuare allo scopo di aumentarne l'efficienza. Il documento prodotto è il nuovo modello delle procedure ottimizzate.

L'attività successiva prevede l'analisi dei requisiti non funzionali: si valutano le necessità dell'azienda in modo da definire i requisiti che il nuovo

sistema dovrà soddisfare. Già si comincia a delineare l'architettura del sistema target in base alle necessità pratiche dell'azienda; questa però, successivamente, ossia in fase di analisi delle soluzioni offerte dal mercato, può subire variazioni poiché le caratteristiche dei prodotti in esame potrebbero portare a rinegoziare i requisiti non funzionali emersi.

In seguito si analizzano le risorse disponibili all'interno dell'azienda, sia in termini di budget che di risorse già a disposizione, ma anche in termini di tempo che è possibile investire nel processo di evoluzione. Dalle valutazioni effettuate si ottiene uno studio di fattibilità che consente di valutare se è possibile o meno proseguire. Anche questo, però, può essere soggetto a variazioni, in base alle considerazioni emerse nella successiva fase al momento della stima dei costi / benefici delle soluzioni in esame.

A questo punto si dovrebbe avere un quadro piuttosto chiaro della situazione in cui si trova ad operare l'impresa e tale conoscenza è anche raccolta, strutturata ed organizzata, nei documenti prodotti nelle fasi precedenti. Si può procedere, dunque, con la fase successiva che prevede la definizione del sistema target. Lo scopo è quello di identificare, fra le soluzioni attualmente esistenti sul mercato, quella che più si avvicina alle esigenze dell'impresa, facendo riferimento al modello delle procedure definito.

La fase di definizione del target consiste nelle seguenti attività:

- 2.1. indagine di mercato dei Sistemi Informativi Aziendali;
- 2.2. selezione delle possibili soluzioni;

poi, per ogni soluzione individuata, si svolgono iterativamente le attività di

seguito,

2.3. live testing;

2.4. stima costi / benefici.

La prima di queste attività prevede un'indagine di mercato volta a produrre un elenco delle possibili soluzioni, sia commerciali che open source, così come è indicato anche nel secondo processo del framework per processi di ESE[19] (Enterprise Systems Engineering) di cui si è parlato nel Paragrafo 2.5, identificando per ognuna di essa, vantaggi e svantaggi. Per classificare i prodotti trovati può essere utile l'utilizzo del modello descritto dal framework basato su MDE[15] (model-driven engineering) descritto nel Paragrafo 2.4.

Fra le soluzioni proposte viene effettuata una prima selezione in modo da ridurre la scelta solo a quei prodotti che sembrano maggiormente applicabili alla realtà aziendale.

Per ognuna delle soluzioni ritenute valide si effettua un'analisi più accurata. Si procede, cioè, con delle installazioni di prova per approfondire e studiare le funzionalità offerte dal prodotto. Questa attività prende il nome di live testing.

Al termine di essa si effettua una stima dei costi da sostenere per l'adozione del prodotto e li si rapporta ai benefici che l'azienda eventualmente ne ricaverebbe dalla sua introduzione.

Termina qui la fase di definizione del target in quanto si hanno a disposizione tutti gli elementi, sia interni che esterni, per poter effettuare la scelta del sistema da utilizzare. Si precisa che questa scelta può consistere

anche in più di una soluzione. Infatti, si può scegliere di integrare le componenti di più sistemi software per soddisfare le proprie necessità.

La fase successiva consiste nell'adeguamento del sistema finale poiché, come detto sopra, spesso non è possibile trovare una soluzione che soddisfi appieno i bisogni dell'impresa.

Di seguito sono elencate le attività presenti nella fase di progettazione:

- 3.1. progettazione componenti software del sistema target;
- 3.2. progettazione delle modifiche al sistema scelto;
- 3.3. progettazione delle modifiche alle procedure ottimizzate.

Queste si occupano da un lato di definire il sistema finale, assemblando le componenti del sistema ovvero dei sistemi scelti nella fase precedente, e dall'altro di progettare un'eventuale adeguamento del prodotto alle procedure interne o viceversa.

Con l'attività di assemblaggio si possono anche includere alcune componenti del sistema attualmente in uso che non siano ritenute obsolete.

Una volta assemblato il sistema ci saranno, probabilmente, delle funzionalità non completamente aderenti alle specifiche aziendali e/o delle funzionalità mancanti. Per tutte quelle procedure che si discostano dalle funzionalità offerte dalle componenti del sistema si può agire in due modi: adeguando le procedure interne in maniera tale da renderle conformi al modo di operare del prodotto, oppure apportando delle modifiche (nel caso ciò sia

possibile¹⁰) al sistema così da renderlo aderente alle procedure. Ovviamente entrambe queste fasi sono opzionali, cioè necessarie solo nel caso in cui si presentino delle non conformità fra le funzionalità offerte dal sistema e le procedure aziendali. Invece, modificare le procedure interne comporta una retroazione alla fase di analisi, e precisamente, alla fase di analisi e prospetto dei miglioramenti, per modificare il modello delle procedure ottimizzate.

Al termine di questa attività si rende necessario, prima di poter procedere, lo sviluppo delle componenti mancanti o da modificare.

La fase successiva, l'integrazione, si occupa infatti di sviluppare quanto progettato nella fase precedente, quindi in base alle modifiche che si è deciso di apportare al sistema scelto.

Questa fase consta di un'unica attività:

4.1. sviluppo / modifica delle componenti software mancanti.

Essa consiste nello sviluppo delle componenti mancanti e/o modifica di quelle esistenti. Ovviamente è anch'essa opzionale secondo quanto stabilito alla fase precedente.

A questo punto l'impresa è in possesso di un sistema target completo, perfettamente in grado di mappare quelle che sono le necessità aziendali. Si può allora procedere con la complessa fase di sostituzione del sistema per introdurre il sistema target all'interno dell'impresa.

¹⁰ Tale scelta è più fattibile nel caso in cui si scelga una soluzione libera o una soluzione commerciale con un produttore che offra un supporto apposito per l'adeguamento del sistema.

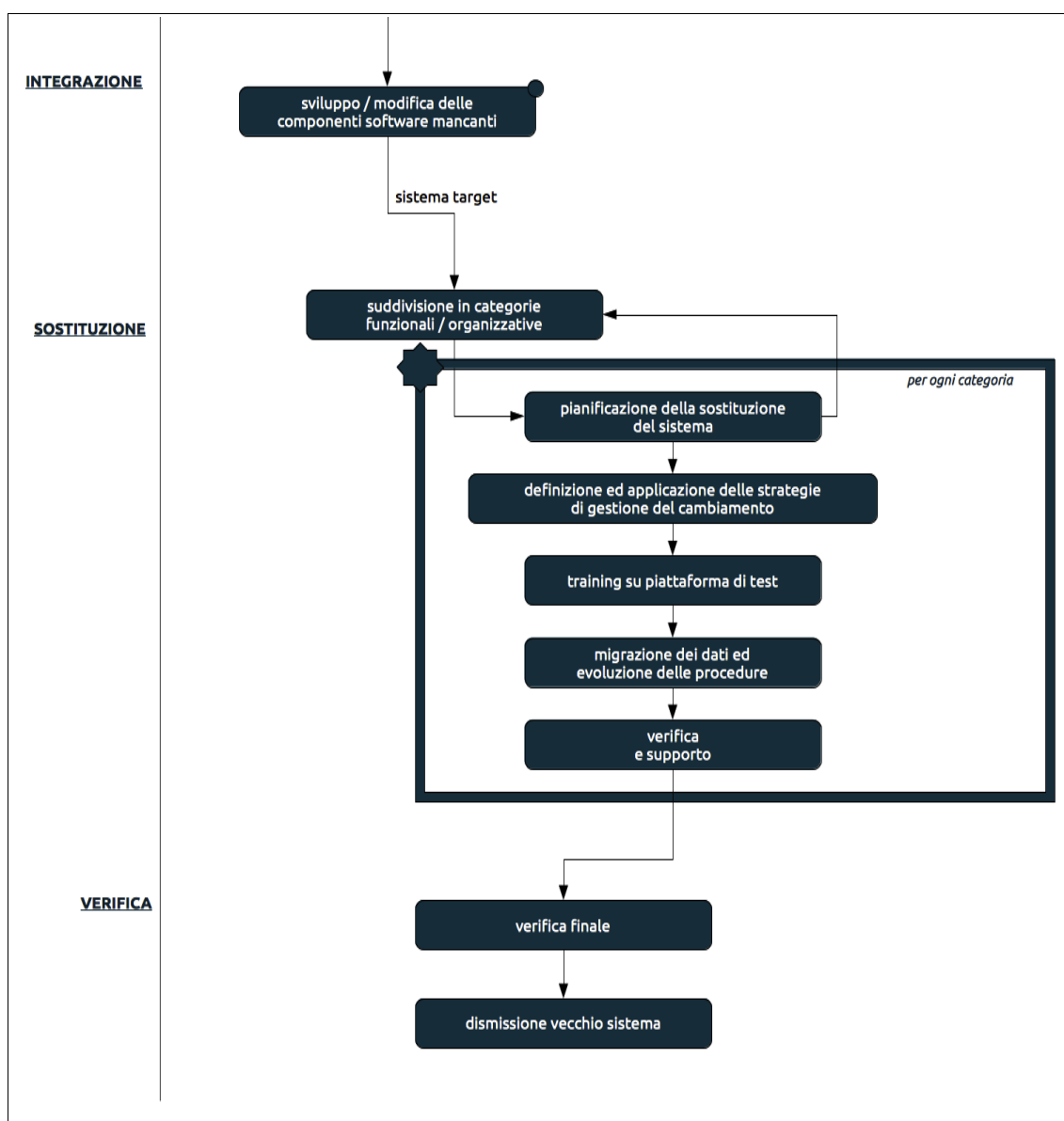


Figura 13: fasi di implementazione del cambiamento (vedi Figura 12)

Le attività di cui si compone detta fase sono:

5.1. suddivisione in categorie funzionali / organizzative;

e per ogni categoria individuata,

5.2. pianificazione dell'evoluzione del sistema;

5.3. definizione ed applicazione delle strategie di gestione del

cambiamento;

- 5.4. training su piattaforma di test;
- 5.5. migrazione dei dati ed evoluzione delle procedure;
- 5.6. verifica e supporto.

L'attività che avvia il processo di sostituzione consiste nel suddividere le componenti in categorie, in modo da procedere poi in maniera incrementale iterando le attività successive previste dalla fase in questione.

Per ogni categoria si pianifica la sostituzione in termini di tempo e attività da svolgere. A questo punto ci si potrebbe rendere conto che la precedente suddivisione in categorie non è adatta, in tal caso è prevista una retroazione all'attività precedente.

Successivamente si definiscono e si applicano le strategie di gestione del cambiamento in modo da ridurre l'impatto interno dell'azienda.

Prima di effettuare la sostituzione vera e propria è necessaria però una formazione degli utenti, a tal fine il framework prevede un'attività di training sulla piattaforma di test.

Quando tutti gli utenti sono opportunamente formati e motivati, grazie alle due attività precedenti, si può procedere con la migrazione dei dati e l'evoluzione della procedure attuando di fatto la sostituzione del sistema in esercizio con il sistema target.

Ultima attività prevista dalla fase di sostituzione del sistema è quella di verifica e supporto. Lo scopo è di verificare che tutto sia andato come

pianificato e supportare gli utenti nell'utilizzo del nuovo sistema.

L'ultima fase prevista dal framework è, poi, la verifica, riferita questa volta all'intero processo. Essa si compone delle attività di:

- 6.1. verifica finale;
- 6.2. dismissione vecchio sistema.

Al termine della verifica, accertatosi che tutto si è svolto come da programma, si procede alla dismissione del vecchio sistema ovvero di quelle parti che non sono entrate a far parte del nuovo.

Nel paragrafo che segue sono descritte in dettaglio le fasi di implementazione del cambiamento. Le prime fasi del framework, invece, sono descritte in [24].

3.3 Le fasi di implementazione del cambiamento

Le ultime tre fasi previste dal framework si occupano dell'implementazione del cambiamento:

4. integrazione del sistema target,
5. migrazione dei dati ed evoluzione delle procedure,
6. verifica della modernizzazione.

Durante le fasi precedenti è stato scelto il sistema target che l'impresa ha deciso di adottare per migliorare il proprio sistema informativo aziendale e le procedure sono state adeguate per essere compatibili con esso. Le fasi di implementazione del cambiamento consentono ora di avviare il processo evolutivo sostituendo di fatto il vecchio sistema e le procedure correntemente in essere. Si arriva così alla prima delle ultime tre fasi previste dal framework con la progettazione delle modifiche da effettuare sul sistema target per terminare l'adeguamento.

3.3.1 Integrazione del sistema target

La fase di integrazione del sistema target, illustrata in Figura 14, consiste di un'attività, resa necessaria da quella precedente di progettazione delle modifiche, che consente di ottenere un sistema target nella sua versione finale.



Figura 14: fase di Integrazione del sistema target (vedi Figura 11)

L'attività prevista, e di conseguenza la fase, diventano superflue nel momento in cui il sistema target aderisce completamente alle proprie procedure, o, grazie alla fase precedente, si è riusciti ad adeguare tutte le proprie procedure e a renderle compatibili con le funzionalità offerte dal

sistema.

3.3.1.1 Sviluppo / modifica delle componenti software mancanti

Quest'attività, schematizzata in Tabella 2, consiste nell'aggiungere le funzionalità mancanti al sistema o nel modificare quelle esistenti non conformi alle procedure attualmente vigenti.

Attività	Sviluppo / modifica delle componenti software mancanti
Fase	Integrazione del sistema target
Input	<ul style="list-style-type: none"> • Progetto delle modifiche <ul style="list-style-type: none"> ◦ Funzionalità nuove ◦ Funzionalità da modificare • Sistema target
Obiettivo	Sviluppo di un sistema target aderente alle procedure aziendali integrando le funzionalità presenti ma non aderenti alle specifiche, e sviluppando quelle mancanti.
Task	<ol style="list-style-type: none"> 1. Individuazione dell'azienda specializzata a cui rivolgersi 2. Definizione della modalità di integrazione
Output	<ul style="list-style-type: none"> • Sistema target

Tabella 2: schema dell'attività Sviluppo / modifica delle componenti software mancanti

Il primo task che l'azienda deve affrontare riguarda la scelta dell'azienda specializzata a cui rivolgersi. Premesso che alcune scelte non sempre sono possibili, poiché dipendono dalla natura libera o proprietaria dei sistemi software adottati, bisogna distinguere due casi:

- funzionalità mancante,
- modifica di una funzionalità esistente.

Nel caso in cui la funzionalità sia completamente da implementare bisogna

scegliere se rivolgersi ad uno dei fornitori dei prodotti software scelti o ad un altro, e se integrare la funzionalità da un punto di vista tecnico in una delle componenti software fra quelle del sistema scelto o realizzare una nuova componente separata da interfacciare poi con il resto del sistema. In alcuni casi la prima scelta, il fornitore a cui rivolgersi, può dipendere dalla seconda, in quanto per la modifica di componenti già esistenti, spesso, è consigliabile rivolgersi al fornitore del componente in questione.

Nel secondo caso, ovviamente, bisogna modificare la componente esistente per sfruttarne le funzionalità già presenti, o al massimo, nel caso in cui non sia possibile intervenire direttamente su di essa, si può costruire un wrapper che consenta comunque di sfruttare le integrazioni della componente con il resto del sistema. In questo caso magari conviene, qualora sia possibile, rivolgersi al fornitore della componente sulla quale si agisce.

Se possibile, conviene sempre integrare le nuove funzionalità con i componenti esistenti. Se i prodotti scelti sono basati su moduli, ad esempio, ed è possibile aggiungerne, è consigliabile svilupparne dei nuovi in modo da garantire in questa fase una migliore integrazione e in futuro una più semplice manutenibilità. Sviluppare una nuova componente isolata dal punto di vista dell'architettura è sconsigliabile, in quanto ciò porterebbe ad una serie di problematiche da affrontare successivamente per garantirne l'integrazione con il resto del sistema.

Per quanto riguarda le modalità di integrazione si possono considerare le seguenti soluzioni:

- lettura / scrittura diretta sull'archivio dei dati;
- comunicazione tramite API / WS (Web Service);
- sviluppo di moduli;
- processi batch.

La prima soluzione risolve la questione dal punto di vista dei dati. Purtroppo, però, essa non è sempre possibile, due le ragioni: in primo luogo si consideri che a volte le applicazioni usano delle basi di dati proprietarie a cui non è possibile accedere, o è sconveniente se ciò risulta troppo complesso in termini di struttura dei dati; in secondo luogo si tenga conto che, solitamente, le scritture dei dati sono complementari ad altre operazioni che, in tal modo, sarebbero da gestire a mano.

La comunicazione tramite API / WS, o quella basata sullo sviluppo di moduli, quando possibili, sono la soluzione migliore poiché entrambe consentono di sfruttare l'interazione con il resto del sistema già prevista in fase di progettazione e sviluppo del prodotto, quindi sicuramente una soluzione più completa e meno rischiosa.

L'ultima soluzione, la comunicazione tramite processi batch, prevede l'esecuzione di script di aggiornamento periodico che consentono di mantenere aggiornati vari archivi di dati. Questa soluzione presenta però diversi svantaggi, è infatti molto facile andare incontro a problemi di inconsistenza dei dati, possono esserci conflitti in fase di aggiornamento e, essendo tale soluzione per sua natura asincrona, può comportare limiti sull'utilizzo del sistema rischiando di lavorare su dati non aggiornati.

Terminata questa fase si ha il sistema finale completo. Il passo successivo consiste nella sostituzione di fatto del vecchio sistema con il nuovo.

3.3.2 Sostituzione del sistema

Questa fase consente la sostituzione del sistema informatico in esercizio in favore del sistema target con la conseguente migrazione dei dati ed evoluzione delle procedure. La fase, le cui attività sono illustrate in Figura 15, si occupa di attuare il cambiamento in maniera incrementale e programmata. Incrementale in quanto si pone l'obiettivo di effettuare la sostituzione per pezzi, e programmata nel senso che definisce accuratamente le singole attività.

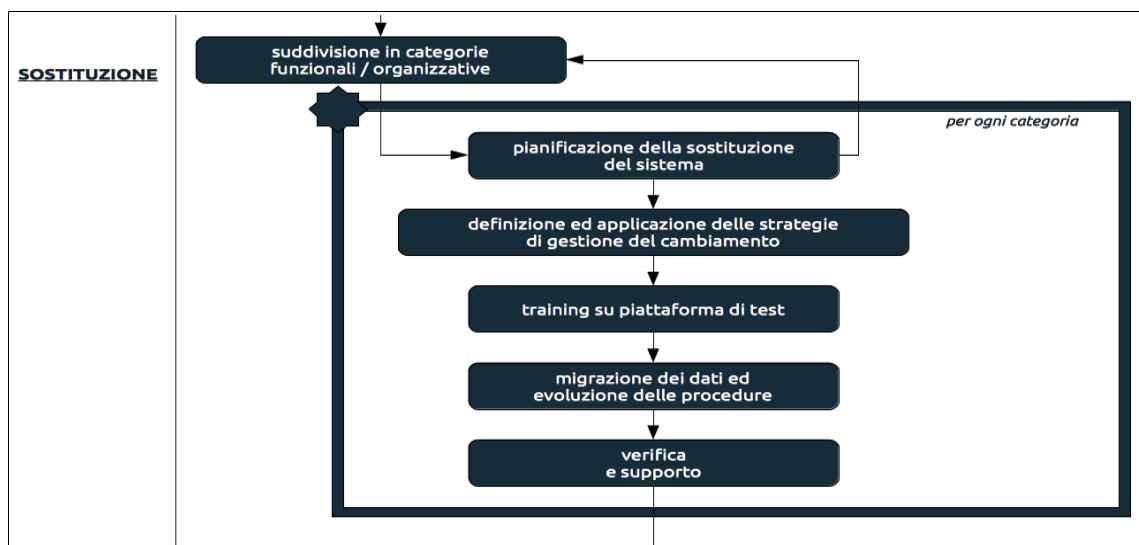


Figura 15: fase di Sostituzione del sistema (vedi Figura 12)

Questa fase riprende le linee guida descritte nel quarto processo del framework basato sulla disciplina ESE[19] (Enterprise Systems Engineering) discusso nel Paragrafo 2.5, esplicitandone però l'incrementalità e la

pianificazione. Inoltre trae spunto anche da alcuni task previsti dal modello per la manutenzione di un ERP[20] studiato nel Paragrafo 2.6.

3.3.2.1 *Suddivisione in categorie funzionali / organizzative*

La prima attività, descritta in Tabella 3, prevede una suddivisione del sistema in categorie funzionali / organizzative, così da poter pianificare l'evoluzione in modo incrementale.

Attività	Suddivisione in categorie funzionali / organizzative
Fase	Sostituzione del sistema
Input	<ul style="list-style-type: none"> • Sistema target • Struttura organizzativa dell'impresa
Obiettivo	Ripartizione del sistema in categorie funzionali tenendo conto della struttura organizzativa interna dell'azienda.
Task	<ol style="list-style-type: none"> 1. Elenco delle funzionalità del sistema 2. Unione delle funzionalità legate da un punto di vista tecnico 3. Unione delle funzionalità che impattano sulla stessa area aziendale
Output	<ul style="list-style-type: none"> • Categorie funzionali / organizzative <ul style="list-style-type: none"> ◦ Funzionalità appartenenti alla categoria <ul style="list-style-type: none"> ▪ Personale addetto alla funzionalità

Tabella 3: schema dell'attività Suddivisione in categorie funzionali / organizzative

La panacea sarebbe quella di suddividere il software in macro-funzionalità e procedere con l'evoluzione in base a tale classificazione. Bisogna però tener conto della struttura organizzativa dell'impresa, specie considerando che una PMI spesso non è caratterizzata dall'averne una struttura con una rigida suddivisione per aree di competenza bensì un'organizzazione interna orientata

ai processi in cui le stesse persone rivestono ruoli diversi. Quindi, conviene tener conto della struttura per effettuare insieme la sostituzione di quelle funzionalità che, anche se slegate da un punto di vista tecnico, impattano sulle stesse persone. Questa scelta permette, ove possibile, di ridurre l'impatto sulle singole persone, ma anche di guadagnarne in termini di efficienza nelle fasi successive dell'evoluzione.

Per effettuare la suddivisione il framework suggerisce un approccio bottom-up. Il primo task prevede di stilare un elenco delle funzionalità del sistema senza considerare la struttura interna e l'accoppiamento fra le componenti. Poi si uniscono quelle funzionalità che sono legate da un punto di vista tecnico, che impattano sugli stessi dati. Le categorie così ottenute, infine, si uniscono ulteriormente tenendo conto di quelle funzionalità che impattano sulla stessa area aziendale.

Nel caso in cui non si riesca ad effettuare una suddivisione completa si possono prevedere dei periodi ibridi in cui si instaurano dei meccanismi di comunicazione provvisori fra il vecchio ed il nuovo sistema in modo da consentire l'aggiornamento dei dati in tempo reale. Ciò può avvenire in quei casi in cui i dati sono fortemente accoppiati¹¹ e non è possibile suddividerli. In questo caso, quindi, il framework suggerisce di installare dei processi di aggiornamento periodico dei dati fra il vecchio ed il nuovo sistema, in modo da consentire l'utilizzo di entrambi i sistemi, il vecchio ed il nuovo, rendendo così possibile la sostituzione del sistema in maniera incrementale.

Fatto salvo che tale suddivisione può essere soggetta a successive

¹¹ In informatica, per accoppiamento o dipendenza si intende il grado con cui ciascuna componente di un programma fa affidamento su ciascuna delle altre componenti.

modifiche con le attività successive si procede, per ognuna delle categorie individuate, con l'evoluzione vera e propria: la sostituzione del sistema e la conseguente evoluzione delle procedure e migrazione dei dati.

3.3.2.2 Pianificazione della sostituzione del sistema

L'attività successiva prevista dal framework prevede la pianificazione della sostituzione del sistema per ciò che riguarda le funzionalità che impattano all'interno della categoria funzionale / organizzativa in questione. Durante questa attività, descritta in Tabella 4, si fa la pianificazione dei passi successivi da compiere e si programmano i tempi, considerando le persone coinvolte e le fasi dell'attività produttiva su cui le funzionalità in sostituzione impattano. È in questa fase, inoltre, che bisogna gestire le funzionalità su cui impattano eventuali procedure ottimizzate in fase di analisi.

Attività	Pianificazione della sostituzione del sistema
Fase	Sostituzione del sistema
Input	<ul style="list-style-type: none">• Vecchio sistema e modello delle procedure ottimizzate• Elenco delle funzionalità
Obiettivo	Pianificare l'evoluzione della singola categoria funzionale / organizzativa
Task	<ol style="list-style-type: none">1. Individuazione della corrispondenza con le vecchia funzionalità2. Gestione dei processi su cui la funzionalità impatta3. Pianificazione dei tempi
Output	<ul style="list-style-type: none">• Corrispondenza con le funzionalità del sistema attuale• Modalità di gestione dei processi in corso• Data presunta della sostituzione

Tabella 4: schema dell'attività Pianificazione della sostituzione del sistema

Il primo task dell'attività di pianificazione consiste, dunque, nell'individuare la corrispondenza tra le funzionalità del sistema vecchio con quelle del sistema target. Le procedure potrebbero essere state ottimizzate in fase di analisi quindi potrebbe non esistere una corrispondenza biunivoca di alcune funzionalità tra i due sistemi. In tal caso bisogna individuare, per ognuna delle funzionalità, quelle che verranno sostituite dalle nuove.

Altro aspetto da considerare, e quindi pianificare, sono i processi in corso. Partendo dal presupposto che lo scopo della metodologia è quello di minimizzare il periodo di inattività questo, seppur breve, non può essere eliminato. Bisogna individuare allora un momento in cui è possibile interrompere l'attività produttiva, effettuare la migrazione dei dati e riprendere i processi congelati sul nuovo sistema.

Durante la pianificazione è possibile che emergano dei problemi relativi alle categorie funzionali / organizzative individuate, nel qual caso è prevista la possibilità di rivedere tale suddivisione tornando alla fase precedente per poi riprendere con le restanti categorie individuate, sempre una per volta.

Delle analisi effettuate è possibile anche stimare una data presunta per la sostituzione.

3.3.2.3 Definizione ed applicazione delle strategie di gestione del cambiamento

La prossima attività del framework, riportata in Tabella 5, prevede la definizione e la successiva applicazione delle strategie da adottare per gestire il

fenomeno della resistenza dei dipendenti.

Attività	Definizione ed applicazione delle strategie di gestione del cambiamento
Fase	Sostituzione del sistema
Input	<ul style="list-style-type: none"> • Elenco delle funzionalità <ul style="list-style-type: none"> ◦ Personale addetto alla funzionalità
Obiettivo	Minimizzare il fenomeno della resistenza del personale.
Task	<ol style="list-style-type: none"> 1. Comunicazione 2. Minimizzazione dell'impatto 3. Coinvolgimento emotivo
Output	<ul style="list-style-type: none"> • Percezione positiva del cambiamento da parte del personale

Tabella 5: schema dell'attività Definizione ed applicazione delle strategie di gestione del cambiamento

Ci si avvale, qui, delle strategie suggerite dal Framework per la gestione del cambiamento[21] descritte nel Paragrafo 2.7. In particolare bisogna individuare le persone su cui l'evoluzione delle funzionalità in esame impatta ed analizzarne gli atteggiamenti, al fine di definire la strategia migliore da adottare.

Per prima cosa bisogna comunicare con le persone coinvolte nel modo più chiaro ed esauriente possibile, illustrando loro i vantaggi introdotti grazie all'adozione del nuovo sistema ed illustrandone le funzionalità, assicurandosi che le abbiano comprese appieno.

In questa fase bisogna, in sostanza, preoccuparsi di ridurre il più possibile l'impatto dell'evoluzione sul personale, questo infatti è una delle principali cause di resistenza al cambiamento.

Per colpire anche la componente conativa degli utenti coinvolti può essere utile individuare quelle persone che rivestono il ruolo di leader naturale

all'interno dell'organizzazione aziendale. Coinvolgere essi per primi consente di influenzare gli altri senza che questi se ne rendano conto. Importante resta il supporto: scopo principale del management è fare in modo che l'utente non avverta la sensazione di abbandono alle prese con il nuovo sistema. All'uopo può essere utile assegnare ad ogni persona un referente, sia esso interno o esterno, a cui gli utilizzatori possono rivolgersi in caso di difficoltà.

Di fondamentale importanza, ancora, è l'analisi del feedback ricevuto dagli utenti che consente di correggere le strategie in corso d'opera. Il management deve essere in continuo contatto con gli utilizzatori del sistema, tramite i referenti o direttamente, a seconda delle dimensioni e della struttura dell'impresa. Il nuovo sistema non deve assolutamente risultare come un'imposizione bensì come un miglioramento dell'attività aziendale. Il management deve mostrarsi disponibile verso i dipendenti, ascoltarne i problemi o le perplessità riguardo l'evoluzione del sistema informativo, e risolvere i problemi da questi sollevati.

3.3.2.4 Training su piattaforma di test

L'attività successiva, illustrata in Tabella 6, consente di massimizzare la conoscenza avviata con la comunicazione nella fase precedente, con il training su una piattaforma di test. Tale fase trae spunto dai task previsti dal modello per la manutenzione di un ERP[20] studiato nel Paragrafo 2.6.

Attività	Training su piattaforma di test
Fase	Sostituzione del sistema
Input	<ul style="list-style-type: none"> • Elenco delle funzionalità <ul style="list-style-type: none"> ◦ Personale addetto alla funzionalità
Obiettivo	Formazione del personale
Task	<ol style="list-style-type: none"> 1. Spiegazione del nuovo sistema 2. Verifica del livello del personale
Output	<ul style="list-style-type: none"> • Consenso a procedere da parte del personale addetto

Tabella 6: schema dell'attività Training su piattaforma di test

Particolarmente utile è effettuare il test su dati reali, magari importandoli ad intervalli regolari dal sistema attualmente in uso. Questa fase serve a garantire l'operatività degli utenti sul nuovo sistema dal momento in cui questo sarà messo in esercizio.

Può essere a tal fine utile coadiuvare l'attività tramite corsi e/o seminari. Resta però di fondamentale importanza supportare gli utenti in ogni momento nell'intento sia di spronarli ad imparare il nuovo sistema che di supportarli in caso di difficoltà, in base anche a quanto detto in fase di definizione ed applicazione delle strategie.

Ad intervalli regolari si verifica il livello di competenza acquisito dai futuri utilizzatori del sistema per poter poi, solo quando questi avranno raggiunto un grado di conoscenza tale da essere operativi, procedere alla fase successiva.

3.3.2.5 Migrazione dei dati ed evoluzione delle procedure

Si procede a questo punto con la sostituzione del sistema, dunque con la migrazione dei dati e l'evoluzione delle procedure, attività riportata in Tabella 7.

Attività	Migrazione dei dati ed evoluzione delle procedure
Fase	Sostituzione del sistema
Input	<ul style="list-style-type: none"> • Sistema target • Sistema attuale • Elenco delle funzionalità • Corrispondenza con le funzionalità del sistema in dismissione • Consenso a procedere da parte del personale addetto • Modalità di gestione dei processi in corso
Obiettivo	Sostituzione delle funzionalità del sistema e migrazione dei dati ed evoluzione delle procedure.
Task	<ol style="list-style-type: none"> 1. Definizione dei tempi per la sostituzione e comunicazione al personale della scelta; 2. Congelamento dei processi in corso; 3. Migrazione dei dati ed evoluzione delle procedure; 4. Ripresa delle attività produttive sul nuovo sistema.
Output	<ul style="list-style-type: none"> • Procedure e dati migrati sul nuovo sistema

Tabella 7: schema dell'attività Migrazione dei dati ed evoluzione delle procedure

È ora che si sfrutta la conoscenza acquisita nelle attività precedenti e si raccolgono tutte le informazioni prodotte. Se le funzionalità in fase di sostituzione riguardano procedure modificate in fase di analisi i dipendenti dovranno aderire alle nuove procedure in base al funzionamento del nuovo sistema. Particolare attenzione va posta alle procedure in corso, infatti, in relazione a quanto deciso nella fase di pianificazione della sostituzione (vedi Paragrafo 3.3.2.2), questa attività va espletata bloccando, seppur per poco tempo, i processi in questione, per poi riprenderne l'esecuzione sulla nuova piattaforma. Questo rappresenta il passaggio più importante e delicato del processo di evoluzione poiché, da questo momento in poi, non è più possibile tornare indietro, a meno di un'ulteriore migrazione dei dati. Infatti dal momento in cui anche un solo processo viene svolto, o ne viene ripresa l'esecuzione di uno già in corso tramite l'ausilio del nuovo sistema, i dati presenti sul vecchio

diventano, di fatto, obsoleti e quindi non più utilizzabili.

Il primo task, per effettuare la sostituzione, consiste nel definirne i tempi. Comunicata la data prevista al personale coinvolto si procede al congelamento dei processi in corso. A questa data si procede con la sostituzione vera e propria migrando i dati sul nuovo sistema di modo che si possano riprendere i processi congelati al task precedente.

3.3.2.6 Verifica e supporto

L'ultima attività consiste nella verifica e nel supporto ed è illustrata in Tabella 8.

Attività	Verifica e supporto
Fase	Sostituzione del sistema
Input	<ul style="list-style-type: none"> • Procedure e dati migrati sul nuovo sistema • Sistema target • Sistema in dismissione • Elenco delle funzionalità • Corrispondenza con le funzionalità del sistema in dismissione
Obiettivo	Verificare l'esito della migrazione
Task	<ol style="list-style-type: none"> 1. Verifica della migrazione 2. Supporto agli utenti 3. Raccolta del feedback
Output	<ul style="list-style-type: none"> • Autorizzazione a procedere alla fase successiva

Tabella 8: schema dell'attività Verifica e supporto

L'attività di verifica ha lo scopo di valutare il risultato della migrazione della singola categoria funzionale / organizzativa verificando che tutto si sia svolto come pianificato e che non ci siano stati problemi.

È previsto ancora il supporto agli utenti. Ciò consente di gestire eventuali problemi non emersi durante la fase di training, e continuare a lavorare sulla componente emotiva degli utenti. È chiaro, infatti, che anche l'utente che non ha avuto particolari problemi se si è sentito supportato durante il passaggio dal vecchio al nuovo sistema sarà maggiormente ben disposto in futuro ad eventuali cambiamenti.

In ultima analisi, in questa fase si raccoglie il feedback ricevuto fin a questo punto su tutto il processo di modernizzazione in modo da tarare la pianificazione delle successive categorie, così come descritto dal quinto processo di ESE[19] illustrato nel Paragrafo 2.5.

3.3.2.7 Iterazione successiva e sviluppo incrementale

Terminata l'evoluzione della singola categoria funzionale / organizzativa si ripetono le fasi descritte, dalla pianificazione alla verifica, in modo da sostituire tutte le componenti del sistema in maniera incrementale, in base a quanto stabilito nella fase di suddivisione in categorie.

L'evoluzione delle singole categorie può essere svolta anche con un discreto livello di parallelismo, in base alla struttura organizzativa e alla capacità del personale, facendo attenzione però ad eventuali vincoli di propedeuticità, tecnica o temporale, che potrebbero esistere fra le funzionalità e/o le procedure appartenenti alle varie categorie.

Terminate le categorie individuate nella prima fase della migrazione si è, di fatto, passati al nuovo sistema.

3.3.3 Verifica della modernizzazione

L'ultima fase prevista dal framework, illustrata in Figura 16, si occupa della verifica del processo di evoluzione.

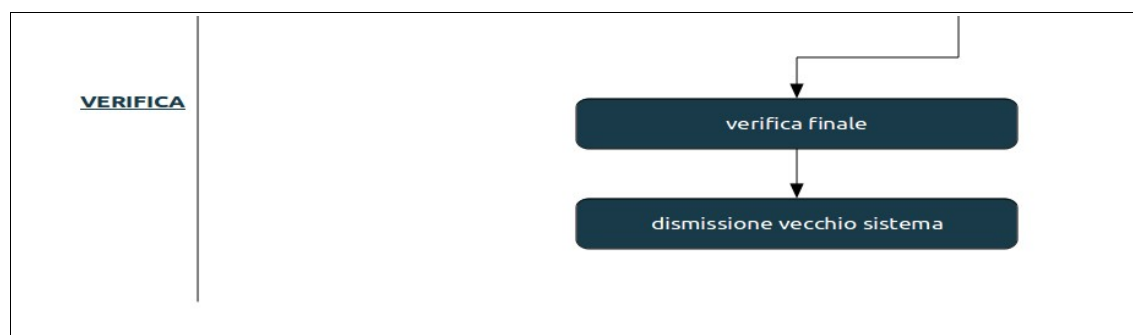


Figura 16: fase di Verifica della modernizzazione (vedi Figura 12)

3.3.3.1 Verifica finale

Si possono a questo punto tirare le somme sull'esito del processo di evoluzione. L'attività di verifica, riportata in Tabella 9, si preoccupa di controllare che tutte le procedure, e quindi le funzionalità, siano state migrate, che i dati siano stati migrati sulla nuova piattaforma e che tutti gli utilizzatori siano operativi sul nuovo sistema.

Attività	Verifica finale
Fase	Verifica della modernizzazione
Input	<ul style="list-style-type: none"> • Sistema in dismissione • Sistema target
Obiettivo	Verificare l'esito definitivo e totale della migrazione
Task	<ol style="list-style-type: none"> 1. Verifica che tutte le procedure ed i dati sono stati migrati 2. Verifica dell'operatività del personale
Output	<ul style="list-style-type: none"> • Esito della modernizzazione

Tabella 9: schema dell'attività Verifica finale

3.3.3.2 Dismissione vecchio sistema

L'ultima attività, schematizzata in Tabella 10, prevista dalla fase, e di conseguenza dal framework, consiste nella dismissione del vecchio sistema ovvero delle parti non riutilizzate. Per vecchio sistema, infatti, si intende l'insieme di quelle parti non riutilizzate nel nuovo sistema. Le componenti del vecchio sistema che si continuano ad utilizzare sono per definizione parte del sistema target.

Attività	Dismissione vecchio sistema
Fase	Verifica della modernizzazione
Input	<ul style="list-style-type: none"> • Vecchio sistema
Obiettivo	Eliminazione del vecchio sistema
Task	<ol style="list-style-type: none"> 1. Spegnimento del vecchio sistema 2. Disinstallazione del vecchio sistema 3. Smaltimento dell'hardware, manuali e tutto quanto relativo ai prodotti non più utilizzati
Output	<ul style="list-style-type: none"> • Terminazione della modernizzazione

Tabella 10: schema dell'attività Dismissione vecchio sistema

3.4 Confronto con le metodologie definite in letteratura

Per concludere riprendiamo le proprietà definite nel capitolo precedente (Paragrafo 2.8), considerate importanti per un framework adatto al contesto di una PMI.

- I. Usabilità
- II. Livello di dettaglio
- III. Supporto tecnico
- IV. Contestualità
- V. Incrementalità

Prima di tutto il framework definito nel presente lavoro è usabile poiché calato direttamente nel contesto di un'azienda.

Fornisce un buon livello di dettaglio in quanto strutturato in fasi ed attività, e per ognuna di esse sono illustrati i task da portare a termine per passare all'attività successiva.

Per quanto riguarda il supporto tecnico ci si avvale degli strumenti già esistenti e comunque si lascia libera l'impresa di adottare formalismi e modelli fra quelli definiti in letteratura.

L'aspetto della contestualità è tenuto fortemente in considerazione in quanto il framework tiene conto dell'intero contesto dell'azienda.

Infine, è garantita l'incrementalità poiché le attività relative alla fase di migrazione sono svolte iterativamente sui componenti del sistema, in modo da

sviluppare il cambiamento in maniera incrementale.

	Usabilità	Livello di dettaglio	Supporto tecnico	Contestualità	Incrementalità
ADM	bassa	alto	alto	bassa	bassa
MDSM	alta	basso	basso	bassa	bassa
Alexandre Cláudio de Almeida et al.	media	alto	medio	bassa	bassa
Robert S. Swarz et Joseph K. DeRosa	alta	medio	basso	alta	media
Celeste See Pui Ng. et al.	alta	medio	basso	bassa	bassa
Adel M. Aladwani	alta	medio	basso	media	Media
<u>Framework definito</u>	alta	alto	medio	alta	alta

Tabella 11: confronto fra il framework definito ed i modelli definiti in letteratura

Si illustrano in Tabella 11 le valutazioni effettuate sul framework appena definito riportando, per un confronto diretto, le valutazioni sulle metodologie analizzate nel capitolo precedente (vedi Paragrafo 2.8) riportate in Tabella 3.

In base a quanto affermato può, dunque, ritenersi il framework particolarmente adatto al contesto delle PMI, applicabile cioè per l'evoluzione dei sistemi informativi aziendali di questa categoria di imprese.

Capitolo 4

Caso di studio

In questo capitolo si presenta un caso di studio reale, ossia quello di un'azienda che si trova ad evolvere il proprio sistema informativo. Lo scopo è l'applicazione del framework definito nel capitolo precedente ad un contesto reale.

4.1 Contesto

La società FRTB¹² (Fabbriche Riunite Torrone di Benevento) è stata fondata il 30 gennaio 1908 su iniziativa dei mastri torronai operanti in territorio sannita ed annovera oggi 8 soci, tutti discendenti diretti del mastro torronaio Mario Rosa.

La società opera in due sedi, come mostrato in Figura 17, ubicate

¹² <http://www.frtb.it> (ultima consultazione novembre 2010)

entrambe a Benevento, che nel seguito identificheremo come Rosa Cash e Fabbriche Riunite.

La sede Rosa Cash si suddivide in tre strutture.

- Il cash è un self-service all'ingrosso specializzato nella vendita di generi alimentari nonché di prodotti non alimentari.
- L'ufficio acquisti è responsabile del ciclo passivo e quindi dell'intero procedimento di acquisto.
- Il CE.DI. (Centro di distribuzione) si occupa della gestione di un gruppo di clienti affiliati ai quali arriva la merce direttamente dal fornitore finale.

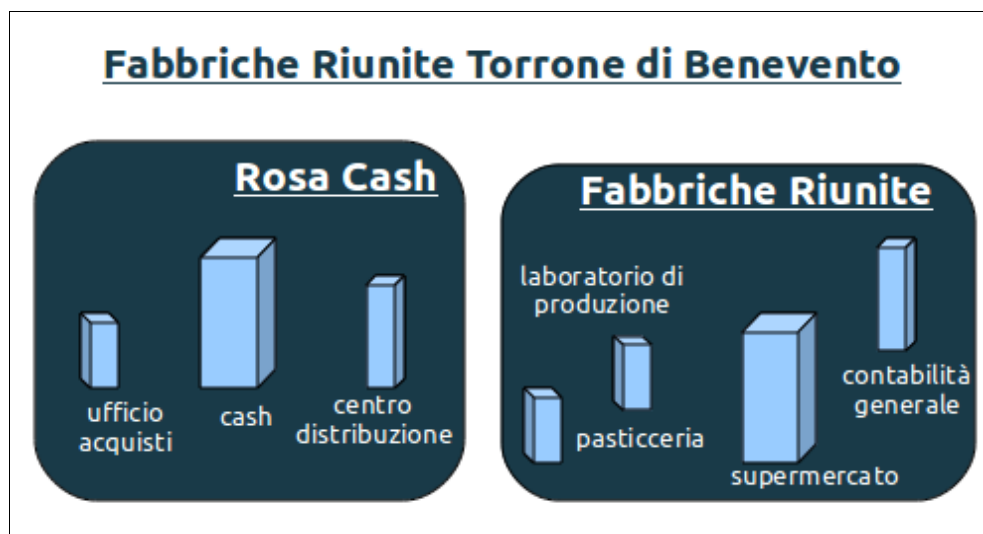


Figura 17: struttura fisica della società Fabbriche Riunite Torrone di Benevento

La sede Fabbriche Riunite presenta invece quattro strutture.

- Il laboratorio di produzione, in cui vengono realizzati i prodotti di pasticceria venduti presso la propria pasticceria, e a terze parti, anche tramite il cash.

- Mario Rosa Pasticciere, dove vengono venduti i prodotti realizzati nel laboratorio, nonché prodotti coloniali.
- Il supermercato, dove si effettua rivendita al dettaglio, i cui prodotti sono acquistati tramite il cash.
- CO.GE. (contabilità generale), uffici di contabilità di tutta la società.

4.2 Architettura del SIA attuale

Il SIA attuale si suddivide in un sistema gestionale, il Food, acquistato da un'azienda di informatica privata, ed un sistema ACG (Applicazioni Contabili Gestionali) di IBM¹³. Entrambi i sistemi, riportati in Figura 18, sono scritti in RPG/400¹⁴, installati su minicomputer IBM AS/400¹⁵ (Application System/400), ed utilizzano come DBMS DB2¹⁶.

13 <http://www.ibm.com> (ultima consultazione novembre 2010)

14 RPG oppure RPG IV è un linguaggio di programmazione nativo per minicomputer IBM della serie iSeries, denominata anche, più comunemente, AS/400.

15 Il sistema AS/400 è un minicomputer sviluppato dall'IBM per usi prevalentemente aziendali, come supporto del sistema informativo gestionale.

16 <http://www-01.ibm.com/software/data/db2> (ultima consultazione novembre 2010)

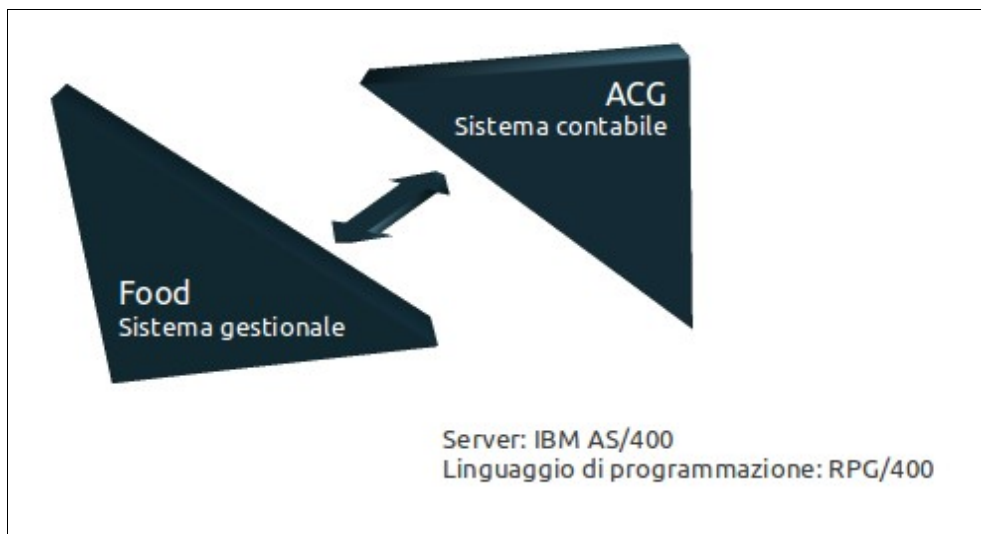


Figura 18: architettura del SIA attuale

Il Food deriva dalla conversione di una precedente versione scritta in RPG II, installata su minicomputer IBM S/36 (System/36)¹⁷, che utilizzava come database dei file, quindi una base di dati di tipo non relazionale. Di conseguenza la versione attuale utilizza solo in minor parte le relazioni, nonostante il DBMS (Database Management System) lo consenta.

La CO.GE. accede all'applicazione ACG, le altre sedi utilizzano invece il Food. ACG consente di gestire tutte le operazioni inerenti gli aspetti contabili mentre il Food supporta l'azienda per le operazioni di acquisto e vendita della merce, carico e scarico del magazzino e per tutto quanto concerne il ciclo attivo e passivo.

Il sistema gestionale Food, in base alle credenziali immesse al login, consente l'accesso solo a determinate operazioni, dunque ogni sede ha le proprie credenziali di accesso.

Presenta, fra le altre, le funzionalità di seguito elencate, secondo quanto

¹⁷ Il sistema S/36 è un minicomputer sviluppato dall'IBM, predecessore del AS/400.

estratto dalla prima schermata presentata dal software a seguito di un accesso con credenziali di utente amministratore.

1. Gestione Anagrafici
2. Gestione Ordini clienti
3. Gestione Documenti Descrittivi
4. Gestione Cassa e Riepiloghi
5. Gestione Ordini Fornitori
6. Gestione Magazzino
7. Gestione Statistiche
8. Gestione Codici a Barre
9. Gestione Promozioni
10. Gestione Inventario/Lifo
11. Gestione Budget/Fatturato Clienti-Fornitori
12. Funzioni di Servizio
13. GESTIONE CONSORZIO TRUST & TRADE

Il software ACG, installato sulla medesima macchina, utilizza lo stesso database, quindi i due software condividono gli stessi dati consentendo così la sincronizzazione.

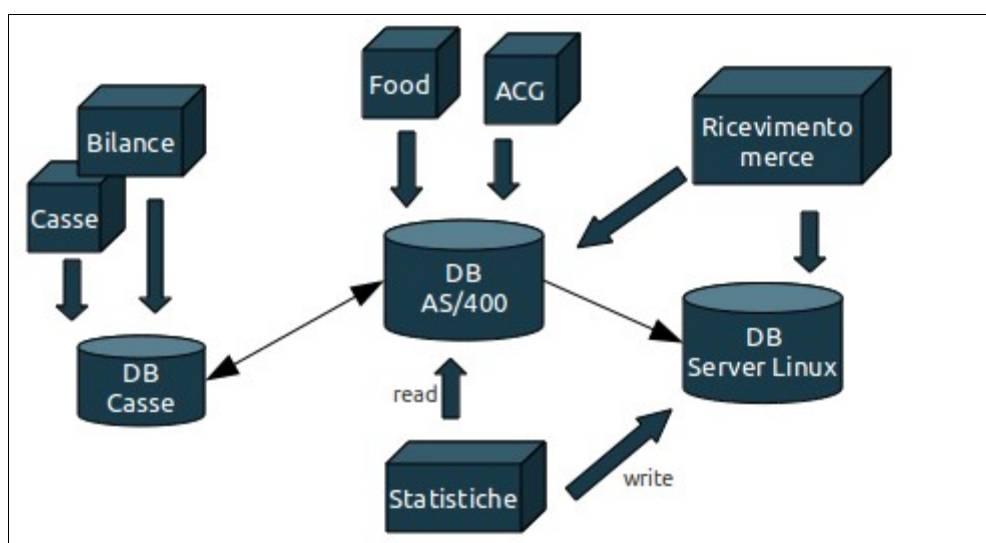


Figura 19: architettura completa del sistema informatico della società Fabbriche Riunite Torrone di Benevento

Il sistema è stato integrato nel tempo con alcune funzionalità sviluppate internamente dalla società utilizzando Ruby¹⁸ come linguaggio di programmazione e il web framework Ruby On Rails¹⁹, PostgreSQL²⁰ e DB2 come DBMS (altre applicazioni sono state realizzate utilizzando Microsoft Visual Studio e Microsoft Access). Tali integrazioni sono dei wrapper del Food e comunicano con esso tramite routine di importazione ed esportazione scritte ad hoc. In Figura 18 si illustra l'architettura completa del SIA.

Le applicazioni esterne, di cui se ne indicano solo alcuni esempi, si appoggiano su un database secondario. Queste applicazioni svolgono una funzione wrapper di alcune funzionalità rendendo possibile la memorizzazione di alcuni dati non previsti dal sistema originario, ovvero incrementando le performance del sistema in lettura. Le applicazioni satellite che fungono da wrapper, quindi, scrivono alcuni dati sul database secondario ed altri sul

18 <http://www.ruby-lang.org> (ultima consultazione novembre 2010)

19 <http://rubyonrails.org> (ultima consultazione novembre 2010)

20 <http://www.postgresql.org> (ultima consultazione novembre 2010)

database DB2 in esecuzione sul sistema AS/400. Ancora, i dati sono scritti direttamente sul sistema AS/400 dalle altre applicazioni e vengono importati sul database secondario con degli script batch in esecuzione periodica. Altre applicazioni si appoggiano a database locali sincronizzati con il database centrale su AS/400.

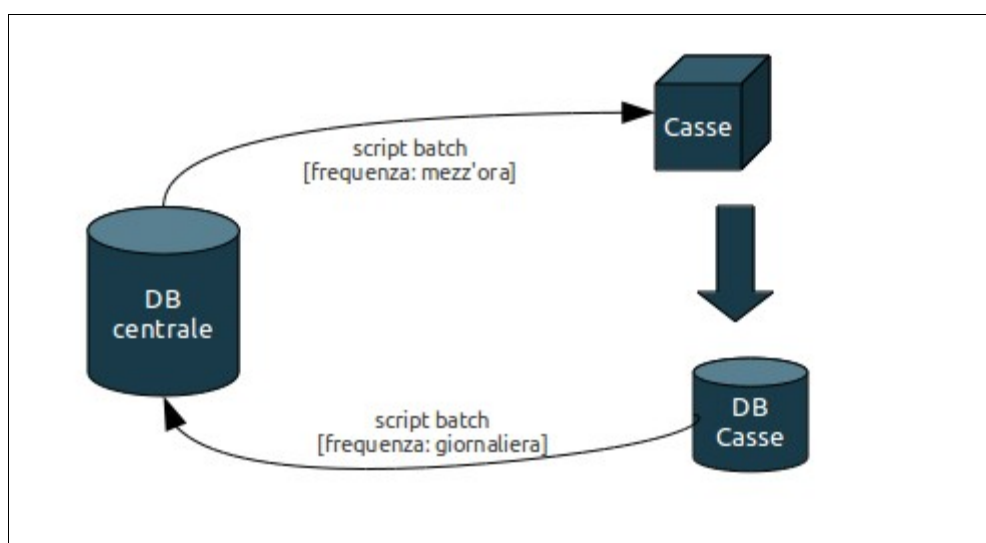


Figura 20: interfacciamento fra il programma Casse e il database centrale (su AS/400)

Di seguito un elenco di alcune applicazioni satellite.

- Casse: il programma consente di gestire le casse del supermercato e si appoggia su database locale. Come illustrato in Figura 20, tale database è costantemente sincronizzato, importando i dati dal DB2 presente su AS/400, mentre i dati generati dal programma sono esportati solo la sera sul database AS/400 tramite esecuzione di processi batch.
- Bilance: il programma gestisce le bilance tracciando i prodotti che vengono pesati e comunicando con le casse di modo che sullo

scontrino fiscale si possa tracciare il prodotto acquistato.

- Ricevimento merce: programma wrapper nato per la gestione delle date di scadenza, dato non previsto dal gestionale Food. Il programma, come schematizzato in Figura 21, scrive sul database DB2 su server Linux e su database DB2 su AS/400; i dati scritti dal Food su AS/400, invece, vengono importati tramite processi batch su db2 su server Linux.
- Statistiche: programma per l'elaborazione di dati statistici. I dati sono letti dal database su AS/400, elaborati per fini statistici e salvati sul database esterno per consultazioni successive.

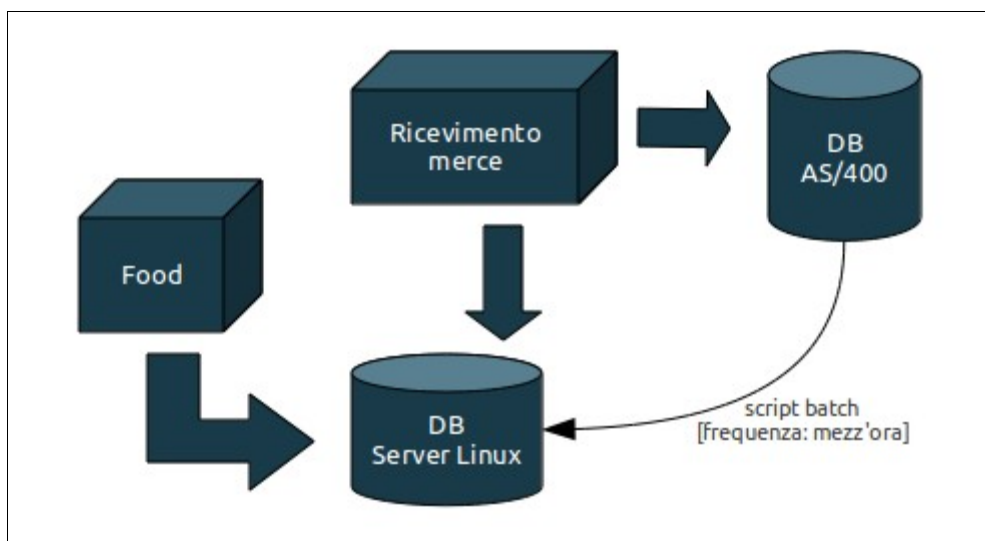


Figura 21: architettura del programma wrapper Ricevimento merce, interazioni con i database

4.3 Applicazione del framework

La necessità dell'evoluzione del SIA della società deriva dalla scadenza della licenza del sistema contabile, nella fattispecie l'applicazione ACG, per la quale IBM non intende garantire più il supporto.

In fase di evoluzione si sceglie di dismettere anche il Food poiché il sistema è legato ad hardware proprietario, il che ne rende più costosa la manutenzione, sia in termini hardware che software, e più difficoltosa l'interoperabilità con l'esterno.

4.3.1 Analisi e progettazione

L'azienda non adotta il framework proposto però è possibile individuare molte similitudini nel processo di evoluzione che la vede protagonista. Di seguito si verifica l'applicabilità della metodologia proposta. Così come è stato fatto nel capitolo precedente, in questa sede si approfondiscono le fasi di implementazione del cambiamento. Per una più dettagliata trattazione delle fasi di analisi e progettazione del cambiamento si rimanda a [24]. Ad ogni modo in Appendice A Procedure aziendali della società FRTB si può trovare una descrizione di alcune delle procedure aziendali.

Dunque il framework non trova una sua completa applicazione nella realtà analizzata, infatti il contesto dell'azienda si diversifica da quella che è la realtà

tipica di una PMI per le motivazioni indicate:

- possiede internamente le competenze informatiche necessarie per l'implementazione di un SIA;
- decide di realizzare una buona parte del proprio sistema target;
- in fase di evoluzione non è prevista alcuna ottimizzazione delle procedure attualmente in essere.

La società possiede internamente una figura EDP (Editing Data Processing) con competenze informatiche tali da essere in grado di implementare i sistemi informatici di cui necessita l'azienda. Infatti, questa decide di sviluppare il sistema gestionale internamente acquistando solo il sistema contabile. La decisione di realizzare un sistema ad hoc deriva dal fatto che i sistemi ERP attualmente presenti sul mercato non sono pensati per realtà che si occupano di vendite all'ingrosso. Per maggiori dettagli sul sistema gestionale si rimanda alla Appendice B Sistema gestionale target della società FRTB.

4.3.2 Implementazione del cambiamento

Le fasi di implementazione del cambiamento previste dal framework sono le seguenti:

4. integrazione del sistema target,
5. sostituzione del sistema,

6. verifica della modernizzazione.

Di seguito si esaminano le attività previste da ogni fase in relazione al caso di studio.

4.3.2.1 Integrazione del sistema target

In questo paragrafo si analizza l'integrazione necessaria fra il sistema gestionale ed il sistema contabile.

Il framework, per questa attività, prevede i seguenti task:

1. individuazione dell'azienda specializzata a cui rivolgersi ;
2. definizione della modalità di integrazione .

A causa della natura chiusa del software acquistato la società è obbligata a rivolgersi direttamente al fornitore del prodotto.

I dati saranno aggiornati tramite routine di importazione ed esportazione dei dati. Il fornitore del sistema contabile collabora con l'azienda al fine di implementare, e quindi aggiungere al prodotto, le routine in grado di importare i dati provenienti dal gestionale ed esportare quelli prodotti dal sistema venduto. Per quanto riguarda le corrispettive routine sul sistema gestionale, ovviamente, la società è in grado di implementare le corrispettive funzionalità. Quanto invece alla modalità di integrazione si implementa una comunicazione tramite processi batch.

4.3.2.2 Sostituzione del sistema

L'evoluzione sarà sviluppata in maniera non incrementale ovvero totale. L'EDP non ritiene possibile suddividere le funzionalità in categorie né tanto meno ritiene necessaria tale suddivisione. Di seguito si procede con l'applicazione del framework considerando tutte le funzionalità come se appartenessero ad una singola categoria funzionale / organizzativa.

L'analisi delle procedure è stata effettuata già in fase di progettazione del sistema e la corrispondenza fra le vecchie e le nuove procedure, almeno per ciò che riguarda il gestionale, è diretta in quanto il nuovo sistema è stato sviluppato in modo da offrire esattamente le stesse funzionalità offerte dal nuovo sistema. La data ultima per l'evoluzione è il 31 dicembre p.v., limite dovuto alla data di scadenza della licenza del sistema ACG. Per quanto concerne i processi in corso l'azienda procederà alla sostituzione del sistema, durante un momento di inattività dei sistemi, quindi in un periodo in cui non c'è alcuna attività produttiva. Infine, quanto al personale coinvolto, non procedendo l'azienda ad un'evoluzione incrementale, esso sarà coinvolto interamente e contemporaneamente anche se per una parte di esso l'impatto sarà minimo in quanto, nonostante l'evoluzione, gran parte delle interfacce di comunicazione²¹ resteranno invariate.

L'attività successiva prevista dal framework prevede la definizione e l'applicazione delle strategie di gestione del cambiamento. L'unica strategia adottata, in accordo con quanto previsto dal framework, è la minimizzazione dell'impatto. Infatti, come già accennato sopra, le funzionalità del nuovo sistema

²¹ Per interfacce di comunicazione si intendono i registratori casse, lettori di codici a barre, POS e quant'altro.

sono state progettate in modo da aderire quanto più possibile a quelle del vecchio sistema.

Maggior peso viene invece data all'attività di training sulla piattaforma di test. L'azienda, sempre in accordo con quanto previsto dal framework, migrerà tutti i dati dal vecchio al nuovo sistema potendo così utilizzare i dati aggiornati per effettuare la formazione sul sistema target; i dipendenti continueranno contemporaneamente ad utilizzare il vecchio sistema. Periodicamente, a valle di una verifica sul livello di comprensione del sistema target, ripeterà l'operazione in modo da effettuare formazione su dati sempre aggiornati.

Quando tutti i dipendenti avranno acquisito le competenze necessarie per utilizzare il nuovo sistema l'azienda effettuerà la migrazione dei dati e di conseguenza l'evoluzione delle procedure sostituendo di fatto il sistema in esercizio, il tutto, ovviamente, durante un periodo di inattività dell'impresa.

A valle della sostituzione l'EDP fornirà il supporto necessario agli utenti ed effettuerà le verifiche di routine per controllare che tutto sia andato correttamente.

4.3.2.3 Verifica della modernizzazione

L'ultima fase del framework prevede la verifica del processo di evoluzione. Nella fattispecie, l'EDP provvederà personalmente alla verifica finale ed alla dismissione del vecchio sistema.

Conclusioni e sviluppi futuri

Nel presente lavoro di tesi è stato definito un framework per la modernizzazione del SIA (Sistema informativo aziendale) di PMI. Il framework consente di guidare un'impresa durante il processo di modernizzazione. Lo scopo non è stato solo quello di evolvere il SIA, bensì di evolvere verso un sistema migliore, ossia sostituire quello attualmente in esercizio con un sistema target in grado di incrementare l'efficienza dell'azienda ottimizzando i processi attualmente in essere.

La prima attività è stata quella di analizzare i framework presenti in letteratura al fine di valutarne l'adeguatezza al contesto delle PMI e al contempo prendere spunto da essi per la definizione del nuovo framework. I framework analizzati, quali per un motivo quali per altri non sono stati ritenuti idonei allo scopo, spesso perché poco concreti e non applicabili o perché concentrati solo su alcune attività dell'intero processo di evoluzione.

In contemporanea all'analisi teorica si è proceduti all'analisi di un caso reale, quello di un'azienda che si trova nella situazione di evolvere il proprio

SIA.

Sulla base della conoscenza acquisita, grazie al lavoro di ricerca ed all'esperienza sul campo, è stato definito un framework pratico e concreto in grado di supportare una PMI durante il complesso processo di evoluzione del proprio SIA. Il framework, grazie anche ad un approccio incrementale, si propone di essere pratico e concreto, quindi, applicabile al contesto reale delle PMI. Esso consiste in una serie di fasi e di attività e supporta l'impresa sia nelle fasi di analisi e progettazione che in quelle di implementazione del cambiamento. Le prime, infatti, consentono di raccogliere informazioni sullo stato dell'arte dell'azienda e, sulla base di esso, di effettuare un'indagine di mercato per identificare, fra le soluzioni attualmente esistenti sul mercato, quella che più si avvicina alle esigenze dell'impresa. Le fasi successive si occupano di adattare il sistema scelto alle procedure aziendali e di sostituire il SIA.

Nell'ultimo capitolo, oltre a presentare l'azienda oggetto del caso di studio, si è verificata l'applicabilità del framework proposto. Di seguito si elencano le lezioni apprese.

- Se non esiste della documentazione aggiornata del proprio SIA è più complicato effettuare la fase di analisi specie se la conoscenza è concentrata in una sola persona.
- Se l'azienda ha al suo interno una figura con competenze informatiche è più semplice compiere il processo di evoluzione.
- Se durante il processo di evoluzione è possibile non modificare le

interfacce di utilizzo del sistema è possibile eseguire l'evoluzione senza preoccuparsi dell'eventuale resistenza da parte dei dipendenti.

Nel presente lavoro ci si è concentrati maggiormente sulle fasi di implementazione del cambiamento, ossia le ultime tre fasi del framework.

Il framework definito espone una serie di passi e linee guida da adottare durante il processo di evoluzione ma per il supporto si avvale di strumenti già esistenti. Un possibile lavoro di approfondimento potrebbe essere, quindi, quello di ampliare gli strumenti da utilizzare come supporto.

Inoltre si potrebbe sottoporre un questionario ad un campione di PMI in modo da capire quali sono i fattori che incoraggiano o scoraggiano l'avvio di un processo di modernizzazione e, per quelle che l'hanno eseguito, quali sono state le attività critiche.

Altro sviluppo futuro potrebbe essere quello di adattare il framework così da gestire quelle situazioni in cui l'azienda decide di realizzare un sistema ERP da zero. La fase di definizione del target potrebbe essere adattata per la ricerca del produttore e, ovviamente, le fasi di adeguamento avrebbero un senso solo per fare leva sui costi.

Infine il framework può essere arricchito anche con un maggiore numero di percorsi alternativi e altre fasi di verifica, con conseguente retroazione alle attività precedenti, per una migliore gestione dei casi di errore.

Appendice A

Procedure aziendali della società FRTB

Per consentire una migliore comprensione del caso di studio in questa appendice si descrivono alcune delle procedure aziendali dell'azienda in oggetto.

Ciclo passivo

Il ciclo passivo è l'insieme di azioni che l'azienda intraprende relativamente alle attività di acquisto di prodotti e per le quali ha delle uscite finanziarie. Rientrano nel ciclo passivo tutte le fasi del processo di acquisto, dal controllo delle scorte agli approvvigionamenti, dalla proposta d'ordine interna alla fattura passiva.

In particolare esso si articola nelle seguenti fasi:

1. controllo scorte;
2. emissione ordine;
3. ricevimento merce;
4. inserimenti dati relativi alla merce ricevuta nel sistema informativo previo controllo di corrispondenza fra ordine, bolla di consegna e merce effettivamente consegnata²²;
5. evasione merce, previo controllo cartaceo, con conseguente carico in magazzino ovvero aggiornamento giacenza;
6. ricezione fattura e pagamento.

Il ciclo passivo ha inizio con l'emissione di un ordine. All'interno dell'azienda ci sono i category manager che, fra le altre cose, si occupano di controllare il livello delle scorte per la propria categoria di competenza. La funzionalità di controllo scorte è presente nel Food e rientra fra le operazioni di "Gestione Statistiche".

A monte di un ordine c'è stata la visita di un fornitore presso l'azienda ed il caricamento dei dati anagrafici relativi agli articoli e ai fornitori nel sistema gestionale Food. Lo stesso articolo può essere associato a più fornitori, quindi il sistema gestisce tale associazione ed il fornitore di default per ogni articolo.

L'inserimento nel sistema dell'ordine fornitore ha come diretta conseguenza la stampa di tre documenti di ordine cartacei: la copia ad uso amministrativo, la copia che verrà consegnata al magazzino e la copia commissione per il fornitore.

²² I controlli sono anche qualitativi, ossia relativi alla data di scadenza, in quanto trattasi di generi alimentari.

A questo punto il magazzino, quando riceve la merce, confronta le bolle di consegna con la sua copia d'ordine. L'addetto utilizza un lettore di codici a barre per caricare direttamente le informazioni relative alla merce ricevuta sul gestionale. Eventuali discrepanze fra l'ordine e la merce effettivamente consegnata sono appuntate a mano sulla copia d'ordine.

L'evasione merce avviene in ufficio acquisti ad opera del personale amministrativo che effettua un secondo controllo cartaceo tra la merce ricevuta e l'ordine emesso. Durante questa fase sono emesse due bolle di consegna, una per la merce ricevuta ed un'altra, eventuale, per il residuo merce ancora da ricevere che sarà inviata in magazzino. È in questa fase che il personale amministrativo può aggiornare l'ordine per correggere eventuali discrepanze tra la merce ordinata e quella effettivamente ricevuta, ovvero per annullare parzialmente l'ordine per la merce ormai ritenuta inutile perché, ad esempio, non consegnata in tempo utile.

Solo a valle di tale controllo avviene l'effettivo aggiornamento delle giacenze di magazzino.

I dati relativi alla fatturazione sono esportati per essere importati nel sistema ACG attraverso il quale procede il ciclo passivo con la ricezione della fattura e fino al pagamento del fornitore.

Ciclo attivo

Il ciclo attivo è l'insieme delle operazioni che un'azienda intrattiene verso i suoi clienti, e che determina i guadagni finanziari per l'azienda, ovvero le operazioni aziendali che permettono di vendere prodotti e servizi nonché di gestire tutte le fasi di vendita.

Il ciclo attivo si articola nelle seguenti fasi:

1. spesa del cliente ed immissione ordine;
2. stampa controllo carrello con conseguente controllo visivo da parte dell'addetto;
3. stampa fattura con conseguente aggiornamento delle giacenze sul sistema informativo;
4. pagamento.

Questo ciclo ha inizio con la spesa del cliente presso il cash. Una volta alla barriera (cassa), vengono caricati sul sistema i prodotti tramite lettore di codice a barre, in modo da comporre l'ordine cliente. Il prezzo risultante per ogni prodotto è dato dai seguenti fattori:

- tipo cliente: personale interno, affiliato o non affiliato;
- prodotti a volantino;
- eventuali costi di trasporto o allestimento, se il cliente ha effettuato l'ordine da remoto e/o se è richiesto il trasporto.

Ad ogni carrello viene stampato un controllo carrello che serve ad effettuare un controllo visivo da parte di un addetto.

Al termine viene stampata la fattura in tre copie, una per il destinatario, una per il mittente (firmata dal destinatario), una per uso interno, e solo eventualmente una quarta per il vettore. All'atto della stampa della fattura avviene l'aggiornamento delle giacenze sul gestionale e l'aggiornamento di un file di accodamento vendite magazzino che permetterà di aggiornare le statistiche tramite routine asincrone.

I dati relativi alla fatturazione sono esportati per essere importati nel sistema ACG attraverso il quale procede il ciclo attivo.

Rifatturazione

Il CE.DI. si occupa anche di gestire i clienti in rifatturazione con il meccanismo della rifatturazione. Questi clienti ricevono la merce direttamente da parte del fornitore della società ma la pagano direttamente a quest'ultima.

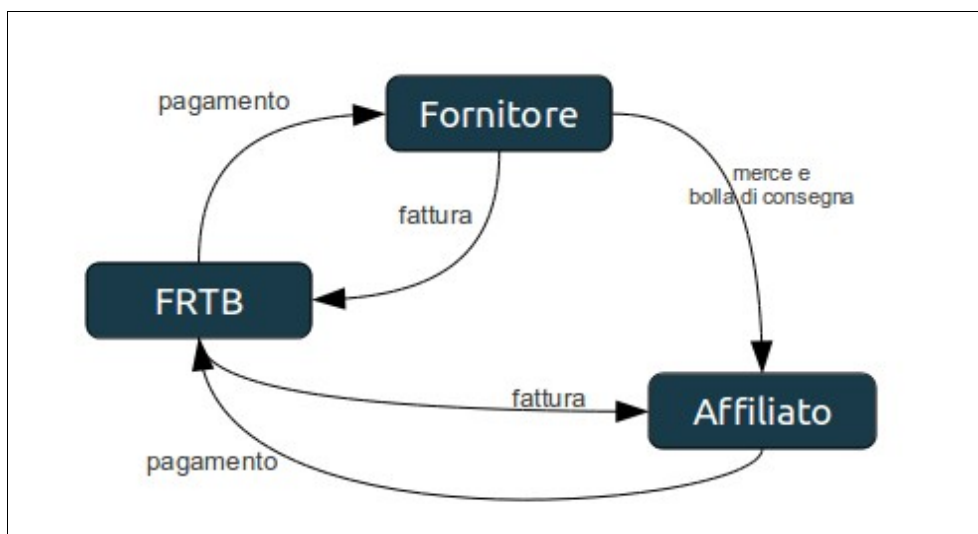


Figura 22: processo di rifatturazione

Il gestionale permette di gestire il processo come descritto in Figura 22.

Inoltre, consente di importare i dati dai file ricevuti direttamente dai fornitori con le informazioni relative alle forniture effettuate per conto della società. Durante questa fase il gestionale controlla le fatture emesse ed i pagamenti ricevuti segnalando eventuali discrepanze.

Rivendite fidelizzate

La società è titolare di due marchi per i supermercati, MAXI PIÙ e MARKET PIÙ, a seconda delle dimensioni. Le rivendite che espongono tale marchio stipulano un contratto in virtù del quale sono obbligate a rifornirsi solo tramite la società.

Le FRTB, periodicamente, prepara un volantino per le rivendite fidelizzate le quali dovranno rivendere i prodotti a volantino ad un prezzo scontato, avendoli esse stesse acquistati ad un prezzo scontato.

Per la gestione delle operazioni ora descritte il Food offre la funzionalità di gestione scontistica accessibile tramite la categoria “Gestione promozioni” in “Gestione Ordini Fornitori”.

Alcune precisazioni

Il Laboratorio utilizza il SIA solo per le operazioni di stampa delle bolle e delle fatture e non per gestire il carico delle merci, ossia delle materie prime necessarie per la realizzazione dei prodotti di pasticceria e rosticceria, in quanto non è di semplice attuazione la corrispondenza tra materie prime e prodotto finale.

Nell'ufficio acquisti, ovviamente, il Food è utilizzato solo per operazioni inerenti il ciclo passivo.

Infine c'è da aggiungere che il Food consente di automatizzare ulteriormente le procedure per quanto riguarda i rapporti fra le sedi, ad esempio per le vendite e gli acquisti fra cash e il supermercato.

Appendice B

Sistema gestionale target della società FRTB

Per consentire una migliore comprensione del caso di studio in questa appendice si descrive il sistema gestionale facente parte del sistema target del processo di evoluzione della società in oggetto.

Architettura

L'architettura del sistema gestionale, descritta in Figura 23, prevede di dislocare i server nelle due sedi operative della società. In ogni sede saranno presenti due o più server sincronizzati tra loro. I server master delle due sedi sono a loro volta sincronizzati.

La presenza di più server in ogni sede garantisce il load-balancing tra i

database e il fault-tolerance in caso di guasto.

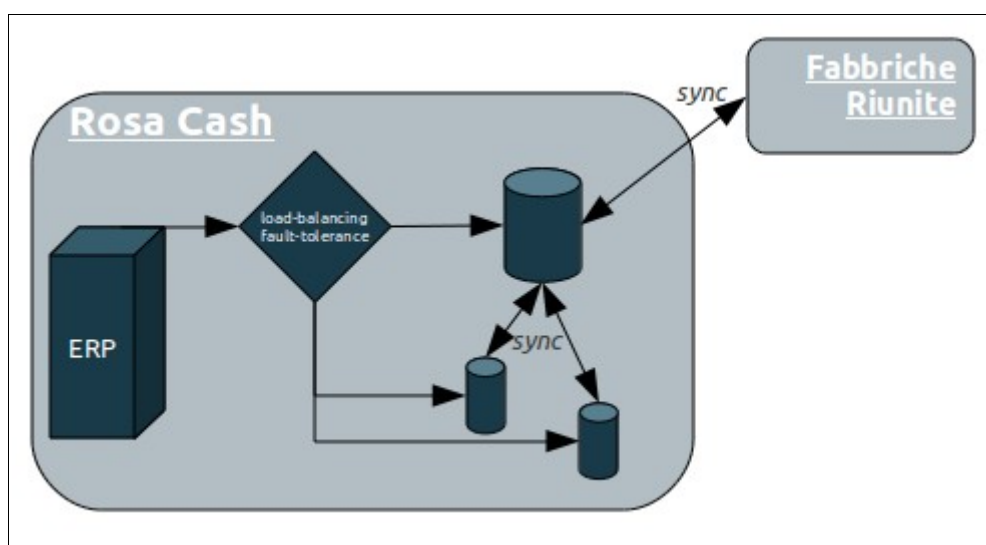


Figura 23: architettura del sistema target

Inoltre, per come è progettata, l'architettura è altamente scalabile, il che significa possibilità di incremento del numero di sedi fisiche, oltre che del numero di database.

Ogni macchina su cui è installato il DBMS ha anche una copia del sistema software, in tal modo, in caso di guasto della macchina che svolge la funzione di server web, gli utenti possono connettersi ad una qualsiasi altra macchina presente all'interno della sede per continuare a svolgere il proprio lavoro.

Il gestionale utilizza una base di dati relazionale, nella fattispecie PostgreSQL, distribuita. Ciò implica meccanismi di sincronizzazione e load-balancing.

Sincronizzazione

Per la sincronizzazione si è scelto di utilizzare RubyRep²³, poiché soddisfa le caratteristiche di seguito descritte.

- Replica multi-master.
- Sincronizzazione asincrona.
- Sincronizzazione a livello di database e non di istruzioni.

RubyRep nasce come soluzione open source per la replica asincrona, master-master di database relazionali, ed ha lo scopo di essere facile da usare ed indipendente dal database. La sincronizzazione avviene tramite dei trigger installati sulle tabelle da sincronizzare; RubyRep controlla ad intervalli regolari se ci sono nuove modifiche da propagare. Il software è scritto in Ruby.

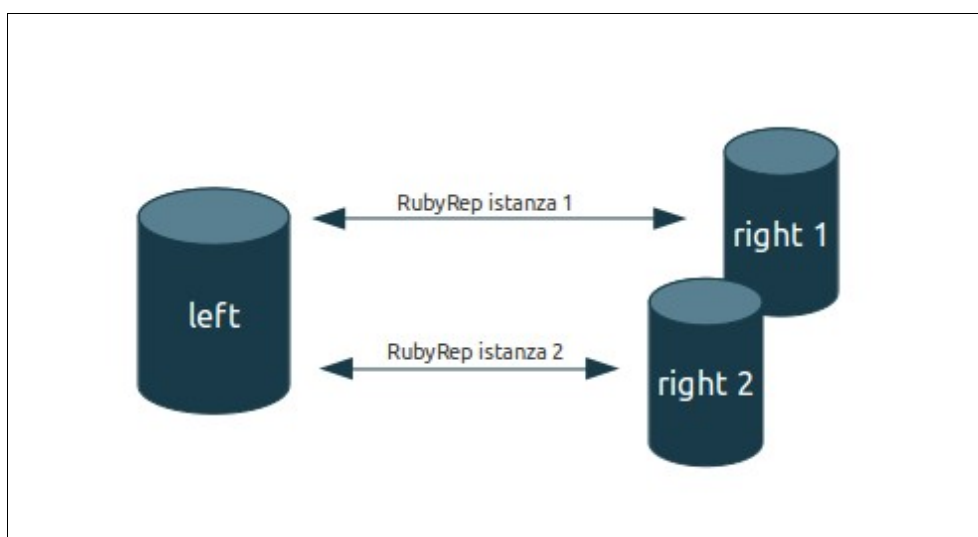


Figura 24: replica multi-master (fra più di due database)

La sincronizzazione avviene fra due master chiamati left e right. Per

²³ <http://www.rubyrep.org> (ultima consultazione novembre 2010)

supportare la sincronizzazione multi-master si adotta una configurazione a stella, mostrata in Figura 24, in cui due istanze di RubyRep agiscono contemporaneamente appoggiandosi sullo stesso database left. Per effettuare questa configurazione bisogna apportare alcune modifiche ai file di configurazione, così da tenere separati le tabelle di appoggio ed i trigger delle due istanze di RubyRep, e modificare ad-hoc i sequence delle varie repliche in modo da evitare conflitti.

Il software consente la gestione dei conflitti oltre che la sincronizzazione delle tabelle del database in cui non sono presenti chiavi primarie, esplicitando i campi da considerare unique nel file di configurazione.

Bibliografia

- 1: Maurizio Pighin, Anna Marzona, Sistemi informativi aziendali, Struttura e applicazioni, Pearson Education Italia, 2005
- 2: Bruce Zhang, History and evolution of ERP, 2005,
http://www.sysoptima.com/erp/history_of_erp.php
- 3: Wikipedia, Enterprise Resource Planning, 2010,
http://it.wikipedia.org/wiki/Enterprise_Resource_Planning
- 4: Canfora, Cimitile, Vetrale, Visaggio, Valutazione empirica dei sistemi ERP Open Source, ConfSL, 2007
- 5: Fabrizio Amarilli, Evoluzione dei sistemi gestionali nelle PMI e nuove modalità di fruizione, Fondazione Politecnico di Milano, 2009
- 6: Franca Cantoni, Gianluigi Mangia, Lo sviluppo dei sistemi informativi nelle organizzazioni. Teoria e casi, Franco Angeli, 2005
- 7: OMG, Object Management Group, 2010, <http://www.omg.org>
- 8: OMG, Architecture Driven Modernization, 2010, <http://adm.omg.org>
- 9: OMG, Model Driven Architecture, 2010, <http://www.omg.org/mda>
- 10: Vitaly Khusidman, William Ulrich, Architecture-Driven Modernization: Transforming the Enterprise, OMG, 2007
- 11: Vitaly Khusidman, Architecture-Driven Modernization and Transformation, OMG, 2008
- 12: OMG, Architecture-Driven Modernization (ADM): Knowledge Discovery Metamodel (KDM), Object Management Group, 2009
- 13: OMG, Architecture-Driven Modernization (ADM): Abstract Syntax Tree Metamodel (ASTM) Specification, Object Management Group, 2008
- 14: Krzysztof Kowalczyk, Model-Driven Software Modernization, Master Thesis in Computer Science, School of Engineering, 2009
- 15: Alexandre Cláudio de Almeida, Glauber Boff, Juliano Lopes de Oliveira, A Framework for Modeling, Building and Maintaining Enterprise Information Systems Software, XIII Brazilian Symposium on Software Engineering, IEEE, 2009
- 16: Douglas C. Schmidt, Model-Driven Engineering, IEEE, 2006

- 17: Wikipedia, Model-driven engineering, 2010, http://en.wikipedia.org/wiki/Model-Driven_Engineering
- 18: Wikipedia, Enterprise systems engineering, 2010, http://en.wikipedia.org/wiki/Enterprise_systems_engineering
- 19: Robert S. Swarz, Joseph K. DeRosa, A Framework for Enterprise Systems Engineering Processes, International Conference on Software and Systems Engineering and their Applications, 2006
- 20: Celeste See Pui Ng, Guy Gable, Taizan Chan, An ERP Maintenance Model, 36th Hawaii International Conference on System Sciences, IEEE, 2003
- 21: Adel M. Aladwani, Change management strategies for successful ERP implementation, Business Process Management Journal, Emerald, 2001
- 22: Sorana Cîmpan, Herve Verjus, Ilham Alloui, Dynamic architecture based evolution of enterprise information systems, International Conference on Enterprise Information Systems, 2007
- 23: Oliver Hummel, Christof Momm, Susan Hickl, Towards quality-aware development and evolution of enterprise information systems, ACM, 2010
- 24: Rosa Principe, Un framework per l'evoluzione dei sistemi informativi aziendali: analisi e progettazione, Tesi di Laurea Specialistica in Ingegneria Informatica, 2010

Indice delle figure

Figura 1: piramide di Anthony.....	13
Figura 2: Relazione tra sistema operativo e sistema informazionale.....	14
Figura 3: evoluzione storica dei SIA.....	17
Figura 4: architetture nei domini di Business e IT.....	33
Figura 5: processi di modernizzazione.....	34
Figura 6: diagramma a ferro di cavallo del processo ADM.....	35
Figura 7: approcci MDSM: RMA e DTA.....	38
Figura 8: architettura delle componenti del framework per modellare, costruire e mantenere SIA.....	40
Figura 9: framework per la gestione del cambiamento.....	48
Figura 10: fasi del framework.....	61
Figura 11: fasi di analisi e progettazione del cambiamento (vedi Figura 12).....	62
Figura 12: legenda schema framework.....	63
Figura 13: fasi di implementazione del cambiamento (vedi Figura 12).....	68
Figura 14: fase di Integrazione del sistema target (vedi Figura 11).....	71
Figura 15: fase di Sostituzione del sistema (vedi Figura 12).....	75
Figura 16: fase di Verifica della modernizzazione (vedi Figura 12).....	86
Figura 17: struttura fisica della società Fabbriche Riunite Torrione di Benevento.....	91
Figura 18: architettura del SIA attuale.....	93
Figura 19: architettura completa del sistema informatico della società Fabbriche Riunite Torrione di Benevento.....	95
Figura 20: interfacciamento fra il programma Casse e il database centrale (su AS/400).....	96
Figura 21: architettura del programma wrapper Ricevimento merce, interazioni con i database.....	97
Figura 22: processo di rifatturazione.....	111
Figura 23: architettura del sistema target.....	114
Figura 24: replica multi-master (fra più di due database).....	115

Indice delle tabelle

Tabella 1: confronto fra i modelli analizzati.....	54
Tabella 2: schema dell'attività Sviluppo / modifica delle componenti software mancanti.....	72
Tabella 3: schema dell'attività Suddivisione in categorie funzionali / organizzative.....	76
Tabella 4: schema dell'attività Pianificazione della sostituzione del sistema.....	78
Tabella 5: schema dell'attività Definizione ed applicazione delle strategie di gestione del cambiamento.....	80
Tabella 6: schema dell'attività Training su piattaforma di test.....	82
Tabella 7: schema dell'attività Migrazione dei dati ed evoluzione delle procedure	83
Tabella 8: schema dell'attività Verifica e supporto.....	84
Tabella 9: schema dell'attività Verifica finale.....	87
Tabella 10: schema dell'attività Dismissione vecchio sistema.....	87
Tabella 11: confronto fra il framework definito ed i modelli definiti in letteratura	89