

## FOCUS

## La gestione della architettura come fattore di riduzione dei costi

La visione indispensabile è quella architettuale: astrarre l'architettura applicativa e tecnologica delle applicazioni e definire un'architettura di base partendo dall'architettura delle singole applicazioni.

**F**ino ad oggi, il primo problema che le aziende hanno dovuto affrontare nell'avviare silos applicativi nuovi, è stato quello di scegliere i prodotti sui quali basare le nuove applicazioni. Questo ha significato l'acquisto di una quantità notevole di middleware, che troppo spesso ha implicato una visione miope dell'investimento: ogni applicazione ha utilizzato il middleware guardando a sé stessa e non all'architettura generale. Ciò ha portato ad avere architetture abbondanti di middleware molto costoso, in molti casi inutile o comunque non ottimale - se visto nella globalità del sistema

informativo -. Tradotto in termini di business significa costi enormi, perché si verifica una moltiplicazione di middleware in produzione, che implica moltiplicazione di canoni di manutenzione e di assistenza di middleware, di tempo da parte degli operatori a rimanere allineati, di upgrade.... La statistica stessa, affermando che molto più del 50% dei budget IT vengono impiegati per attività di questo tipo, evidenzia uno stato dei sistemi informativi troppo spesso ridondante e con margini di miglioramento molto alti.

La strada che crediamo possa portare ad un'evoluzione positiva dei si-

stemi informativi è quella dell'individuazione di un'architettura di riferimento, guardando aldilà delle singole applicazioni, mettendo in discussione, quando necessario, parti di middleware che sono o ridondanti o rispetto all'altro, oppure che vengono sfruttati solo per alcune loro funzionalità.

La visione che riteniamo sia indispensabile oggi è la visione architettuale, cioè la capacità di astrarre l'architettura applicativa e tecnologica delle applicazioni ed avere la capacità di definire un'architettura partendo dall'architettura delle singole applicazioni. Vanno sostenute manutenzioni evolutive del sistema, non più legate all'evoluzione di una singola applicazione, ma connesse all'evoluzione di tutto il sistema informativo.

Inoltre, una simile gestione dell'architettura ha un impatto economico sull'IT management enorme: possono risultare risparmi su questi costi che si aggira-

no tra il 50 ed il 60% nell'arco di un anno. Ridurre i costi di manutenzione ed i canoni, in una visione di business, significa risparmiare più soldi del budget per la manutenzione utilizzandoli, invece, per investire nel miglioramento che avviene nell'aver sistemi in grado di fare le cose più velocemente, con progetti che costano meno anche nella loro fase di costruzione. Caratteristiche, queste, che non sono possibili con un sistema informativo carico di componenti pesantissimi, costosissimi, e che troppo spesso creano un "locking" sul fornitore...

Proporre questo tipo di gestione dell'architettura è un chiaro segnale di evoluzione e maggiore maturità del mercato. Quando alcuni anni fa si trattava di investire in nuove tecnologie, era evidente che la scelta di un cliente si indirizzasse verso quei fornitori in grado di offrire maggiori garanzie e soprattutto quelli che,

### SOMMARIO:

- › *FOCUS*  
La **gestione** dell'**architettura** come fattore di **riduzione dei costi**
- › *PROJECT MANAGEMENT*  
Il **costo** della "**non qualità**"
- › *TECNOLOGIE*  
La **gestione** degli **ambienti**
- › *NOTIZIE DA MOKABYTE.it*

**CASE STUDY**

*Sotto ad un'applicazione è stato installato un integration broker che ha anche funzioni di workflow, perché serviva l'integration broker, mentre sotto ad un'altra applicazione è stato installato un diverso motore di workflow perché in un altro progetto non serviva un integration broker. Questo significa che alla fine dei due progetti nel sistema sono installati un integration broker con workflow di cui non viene usato il workflow, ed un'altra applicazione che ha un motore*

*di workflow diverso dal primo. I due middleware vanno: mantenuti, cambiano di versione ogni anno, vanno mantenute persone con skills specifiche, vanno pagati upgrade, canoni di manutenzione, vanno fatte nuove andate in produzione, nuovi test ad ognuna di queste versioni nuove. In realtà questo potrebbe essere ridotto compatando tutto su un motore solo, utilizzando un solo middleware per le due funzioni.*

per solidità della propria struttura, si pensava fossero in condizioni di far funzionare al meglio questi middleware. La storia recente dell'IT ha, invece, dimostrato che troppo spesso le condizioni eco-

nomiche per far funzionare al meglio silos applicativi stanno impiegando una grande fetta dei budget dei clienti. L'esperienza di alcuni anni fa era diversa rispetto a quella attuale e la stessa matu-

rità nella scelta delle forniture è cambiata. In questa fase è importante non guardare indietro, non ritenere errori di valutazione quelle scelte che oggi stanno dimostrandosi dispendiose. Vanno attivati, invece, quei processi per ridefinire le architetture dei singoli middleware con uno sguardo globale, rivolto all'architettura di riferimento dell'intero sistema informativo. Una volta che la scelta su una particolare tecnologia è stata presa, è necessario lavorare di affinamento, avendo una serie di strumenti e di esperienza in più rispetto a quando avevo fatto la scelta. Inoltre, dei costi di manutenzione, assistenza e upgrade che

stanno venendo alla luce adesso, non se ne aveva la cognizione. A questo punto, però, avendo maggiore esperienza e padronanza di queste tecnologie si può decidere di fare interventi architetturali per ristrutturare anche quei costi. Gestire e ristrutturare l'architettura secondo questa visione, si evidenziano non solo dei decisivi risparmi sui costi, ma miglioramenti in termini di performance, di pesantezza – sui tempi di intervento e su quelli di creazione di nuove applicazioni -, di agilità della gestione, leggerezza dell'architettura e quindi velocità. ■

**PROJECT MANAGEMENT****Il costo della "non qualità"**

Non affrontare in maniera rigorosa requisiti di qualità nella fase iniziale di un progetto, può portare ad un aumento dei costi stimabile tra il 20 e il 40%...

**P**er qualità intendiamo tutti quei particolari che sono presenti in un progetto e che garantiscono la gestione di una buona scrittura del codice. Questi particolari sono requisiti indispensabili di qualità e possiamo evidenziarli come: rigore nella definizione dei requisiti, definizione a monte del progetto di un'architettura applicativa, creazione di linee guida per gli sviluppatori, automazione dei test e definizione rigorosa del

change management, la gestione dei cambiamenti che devono essere fatti. Stabilire un costo della non qualità significa possedere gli elementi capaci di comprendere che cosa vuol dire non avere rigore sui progetti. Non avere rigore significa fare progetti in cui ci si preoccupa più di mandare in produzione un risultato più che preoccuparsi della qualità del risultato creato. Sono nate una serie di discipline importanti che tendono a raggiungere un processo

di costruzione del software in scala industriale, proseguendo verso una progressiva maturazione del settore che ancora in molti casi si presenta come artigianale. La gestione del progetto sta e deve assumere nuovi connotati, ma prima di tutto deve rispondere a particolari specifiche di qualità e rigore dell'affrontare qualsiasi processo. Questa è una strada obbligata anche dal punto di vista economico. In molti casi abbiamo misurato, a po-

steriori, il costo di non aver rispettato requisiti di qualità lungo il progetto. Nella casistica di cui siamo in possesso, il non affrontare in maniera rigorosa questi requisiti in fase iniziale di progetto, ha portato ad un aumento dei costi che si aggira tra il 20 e il 40%. Questi dati sono supportati anche dalla letteratura. Basta prendere tutte le tecniche di stima attualmente in uso e si può vedere quanto incide non avere questo livello di qualità.

Non rispettare questi requisiti, inoltre, spesso significa non raggiungere nemmeno la qualità attesa. È fondamentale evitare degenerazioni generate proprio da una mancanza di rigidità ad inizio pro-

getto. Nella letteratura che viene scritta dagli esperti del processo di sviluppo, le stesse tecniche agili di sviluppo dicono di fare la documentazione base che serve, non dicono il contrario. Oggi in tutti i pro-

getti è possibile misurare il costo della non qualità nell'esecuzione. Ci sono gli spazi per migliorare qualità e al tempo stesso abbassare i costi di sviluppo, anche se va tenuto ben presente che un buon

processo non si compra e nemmeno gli strumenti possono garantire risultati miracolosi, ma solo supportare un processo. ■

## TECNOLOGIE

### La gestione degli ambienti

Un'infrastruttura che fornisce funzionalità di management e sicurezza per mantenere indipendenza tra fornitore e consumatore del servizio

La gestione degli ambienti è una questione strettamente correlata con il cambio delle tecnologie in atto. Dai singoli silos applicativi che venivano installati, si sta passando verso architetture sempre più orizzontali nelle quali risulta improbabile pensare che questi singoli applicativi possano essere indipendenti. Lo stesso passaggio che si è verificato con J2EE ha messo in luce questo cambiamento, mettendo in comunicazione tra loro più silos applicativi attraverso l'application server. Di fatto, oggi, due applicazioni J2EE condividono una parte dell'applicazio-

ne che sta nell'application server. Questo ha provocato maggiore difficoltà nel mantenere operativi gli ambienti, proprio perché queste singole applicazioni non risultano più isolate tra loro. Inoltre, la questione è resa più complessa dalla presenza di un numero di ambienti diversi: uno di sviluppo, uno di test, uno di collaudo o pre-produzione e un altro di produzione, per citarne alcuni. Tenere sotto controllo gli ambienti, sia dal punto di vista operativo che di costi, diventa sempre più un problema. Un singolo ambiente per di più costa centinaia di migliaia di euro, quindi

fare un errore anche su uno solo di questi ambienti diventa molto costoso. Se, per la non chiarezza di ciò che mi serve, spreco o non sono efficiente su un ambiente rischio di buttare centinaia di migliaia di euro.

La nuova frontiera sul tema della gestione degli ambienti è data dalla capacità di definire questi ambienti nel modo migliore possibile e la capacità di tenerli sotto controllo. Nel mercato attuale non ci sono strumenti affermati che consentono di controllare con precisione sia la parte hardware che la parte software di base che quella applicativa. Inoltre,

troppo spesso queste due ultime parti sono confuse o mischiate. Gli strumenti tradizionali non sono sufficienti perché manca completamente la possibilità di monitorare efficacemente la parte applicativa, per di più non c'è ancora una cultura da parte di chi scrive applicativi che consenta di facilitarne il monitoraggio.

Quindi si afferma che il problema della gestione degli ambienti si debba affrontare contemporaneamente sia sul fronte degli strumenti sia sul versante delle modifiche che riguardano l'architettura e al design delle applicazioni. ■

#### COMUNICATO "Offshoring: siglato l'accordo Imola - Rave Technologies"

Siglato l'accordo di partnership tra **Imola** e la società indiana **Rave Technologies** per l'affidamento in offshoring dei processi di sviluppo software dei progetti commissionati da Clienti italiani a Imola. Imola, che nel 2004 ha registrato alti tassi di crescita nel settore della consulenza IT, ha già avviato due progetti presso propri Clienti, a conferma della fiducia del mercato nella scelta

imprenditoriale fortemente voluta dall'amministratore di Imola, **Claudio Bergamini**. «L'accordo di partnership con Rave Technologies di Mumbai (Bombay), India – spiega **Claudio Bergamini** – può offrire una grande complementarità all'attuale offerta di Imola. Rave Technologies ha dimostrato di avere grande attenzione alla qualità del processo, caratteristica che per

Imola è fondamentale nella propria mission. Inoltre, i quindici anni di esperienza in offshoring dell'attività di sviluppo software, unita ad uno staff molto qualificato che garantisce grande scalabilità, sono gli ingredienti che abbiamo ritenuto fondamentali nella scelta di Rave Technologies come nostro partner. Questo accordo permetterà di proporsi sul mercato dei progetti mission

critical con uno staff misto composto da professionisti di Imola nei ruoli ad alto profilo e di riferimento al Cliente, e con la maggior parte dei ruoli di sviluppatore affidati a Rave Technologies. Vogliamo mantenere la professionalità e serietà conquistata da Imola nel corso di oltre 20 anni di progetti presso Clienti, ma con la possibilità di presentarci con un buon rapporto qualità/prezzo».

**NOTIZIE da MOKABYTE.it**

**FORMAZIONE** dalla redazione MokaByte

**CALENDARIO DEI CORSI 2005 / 2006**

(per maggiori informazioni [www.mokabyte.it](http://www.mokabyte.it))

Percorsi formativi	gg	2005					2006					
		sett	ott	nov	dic	gen	feb	mar	apr	mag	giu	
Formazione in aula	Architettura J2EE	2-3			▶					▶	▶	
	Hibernate	2		▶								
	Java OOP	3							▶			
	JSF	2						▶				
	J2EE EJB	3			▶							
	J2EE WEB	2		▶								
	Struts	2							▶			
	Java Web Services	2							▶			
	UML	3							▶	▶	▶	
Workshop	J2EE EJB + WEB deployment con JBoss	3		▶								
	Architetture J2EE dalla progettazione all'integrazione	3		▶	▶							

**Legenda:**

▶	Si rimanda l'approfondimento di Struts al corso specifico	▶	Corso: illustrazione esercitazioni con supporto dedicato per le esercitazioni su forum
▶	Corso pratico	▶	Presentazione in anteprima del libro "Architettura J2EE, dalla progettazione all'integrazione" scritto dagli stessi docenti/conducenti del workshop

gruppoimola [ news ]

Sede Legale  
 Imola Informatica s.r.l.  
 Via Selice, 66/A  
 40026 Imola

Redazione  
 Nicolai Studio  
 Via della Valverde, 32  
 37122 Verona

Contatti Redazione  
 Gabriele Nicolis - Ruth Humphries  
 email > [info@nicolaistudio.it](mailto:info@nicolaistudio.it)  
 tel. > 045.8036.841  
 fax > 045.8026.848

Policy  
 Il materiale di questa newsletter è di proprietà intellettuale del Gruppo Imola (Imola Informatica srl & MokaByte srl). MokaByte® è un marchio registrato da MokaByte srl. Java®, Jini® e tutti i nomi derivati sono marchi registrati da Sun Microsystems. Tutti i diritti sono riservati. E' vietata la riproduzione anche parziale del documento previa autorizzazione. Per eventuali comunicazioni inviare un email a [info@nicolaistudio.it](mailto:info@nicolaistudio.it)

**REPORT** dalla redazione MokaByte

**OMG Workshop: MDA, SOA e Web Services**

Ci teniamo a condividere alcune impressioni scaturite dalla nostra partecipazione al seminario di Orlando, Florida, dal titolo **"Delivering the Integrated Enterprise - Practice, not Promise"**, che ci ha premesso di verificare lo stato dell'arte di SOA (Service Oriented Architecture), di MDA (Model Driven Architecture) e la maturità tecnologica dei Web Services.

**SOA.** Data l'ampiezza del tema, è difficile formulare una definizione esauriente e riconosciuta di SOA e anche nell'ambito della conferenza si è molto insistito su quello che SOA non è (es. non è una tecnologia, non sono Web Services, non è un prodotto, non è una Metodologia, non è un Processo, non è una piattaforma definita, ...).

Invece, come approccio architeturale, SOA investe tutti gli aspetti associati alla definizione di un Information System (metodologici, di processo, organizzativi ed anche tecnologici), ma soprattutto si pone l'obiettivo di guidare la definizione di un'Architettura Enterprise ispirata e modellata in funzione dei driver di Business. SOA non propone, quindi, concetti architettureali "nuovi", quanto una visione integrata ed ampliata di principi architettureali già adottati nel passato (es: CORBA e J2EE) e di problematiche affrontate in altri ambiti (es. EAI e BPM). SOA è un insieme di principi, linee guida, best practices architettureali indipendenti da qualsiasi tecnologia.

**Web Services.** Per quanto detto sopra, è evidente come il tema SOA sia solo in minima parte tecnologico, ma è altrettanto evidente come

esistano tecnologie abilitanti (es. CORBA, J2EE e .NET) o che comunque meglio si prestano all'implementazione dei suoi principi di progettazione. Parere condiviso e ribadito più volte nel seminario, è che in ambiti SOA sia particolarmente interessante lo stack tecnologico dei Web Services.

**MDA.** È un approccio per la progettazione e lo sviluppo di software definito da Object Management Group ed adottato da numerosi vendor. La specifica OMG incorpora una serie di standard come UML, Meta-Object Facility (MOF), XML, Meta-Data Interchange (XMI) ed il Common Warehouse Meta-model (CWM).

[...]  
 Ci riserviamo la pubblicazione di un report più completo nel prossimo numero della newsletter.

Claudio Bergamini & Stefano Rossini