**BOAT**

Un modo di scomporre l’Enterprise Architecture è attraverso il sistema BOAT, che scompone la EA nei seguenti livelli:

- B: *Business*

- A: aspetti *Architetturali* = Architettura funzionale (= parti software)

- O : aspetti *Organizzativi*

- T : aspetti *Tecnologici*

**VISTA DI BUSINESS**

Analisi

La vista di business può essere analizzata attraverso tre elementi:

1. Partecipanti
2. Oggetti dell’interazione
3. Orizzonte temporale

1) *Partecipanti*:

I partecipanti sono le diverse tipologie di utenti del SI.

Si classificano come:

* Business (B), cioè aziende, se il SI è rivolto ad azienda
* Government (G), cioè pubbliche amministrazioni
* Citizen (C), cioè cittadini, consumatori

Il sistema è B2B se riguarda l’interazione fra aziende, B2C se riguarda l’interazione consumatore-azienda, e via così.

2) *Oggetti dell’interazione*:

Gli oggetti dell’interazione, cioè ciò che viene “scambiato” dagli utenti, possono essere:

* Prodotti fisici
* Servizi
* Prodotti digitali
* Prodotti finanziari
* Oggetti ibridi, cioè oggetti che appartengono a più di una delle quattro categorie elencate sopra.

3) *Orizzonte temporale dell’interazione*:

A seconda della stabilità dell’interazione (un utente si rivolge sempre a un altro, oppure solo in alcuni casi, oppure a seconda della situazione, eccetera…), l’interazione può essere classificata dal punto di vista temporale come:

* *Stabile* = il partecipante si rivolge sempre a un dato altro partecipante, la relazione è costante nel tempo e non subisce modifiche.
* *Semi-stabile* = La relazione è grossomodo costante, con qualche variazione ogni tanto
* *Dinamica*: la relazione è limitata a un singolo ordine, a una singola transazione.
* *Ultra-dinamica:* la relazione può cambiare anche all’interno della transazione, cioè durante la transazione se c’è un problema posso decidere di cambiare partecipante con cui sto interagendo ed eseguire la transazione con un altro partecipante.

Business driver

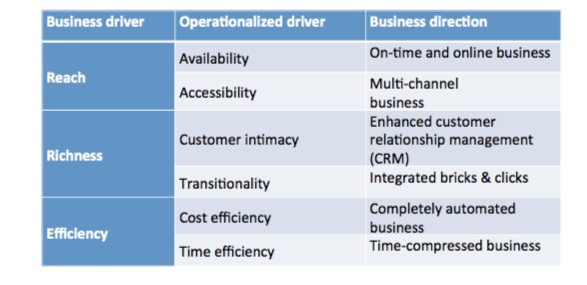
I business driver *sono i fattori che motivano il progetto del SI (o la sua evoluzione) identificano gli obiettivi dell’intervento*.

Bisogna rendere questi obiettivi misurabili e per farlo si definiscono tre tipi di variabili: input (= risorse necessaria per svolgere attività), output atteso (=risultato desiderato) e output effettivo (=risultato ottenuto).

I driver possono essere di vari tipi:

1. *Efficacia*: è la capacità di raggiungere gli obiettivi previsti, definita come **output-effettivo/out-put atteso**.
2. *Estensione*: consiste nell’ampliare l’ambito di attività di un’azienda, ad esempio estendendosi sul territorio, aumentare ore di attività, il numero di canali usati per interazione con gli utenti.
3. *Ricchezza*: è l’intensità di comunicazione fra azienda e utenti.
4. *Efficienza*: si misura in **output-effettivo/input**, cioè quanto bene si riesce a sfruttare le risorse impiegate. L’efficienza può essere quantificata anche con la piramide di Anthony: risparmiare risorse (umane, ad esempio) significa ad esempio ridurre il personale impiegato nelle attività di routine, riducendo quindi il livello operativo alla dimensione di quello strategico (quindi passiamo da una piramide a un rombo).

Gli operationalized driver sono i driver effettivi, quelli messi in pratica (= cioè concretamente come realizziamo i driver elencati qua sopra).



L’estensione è realizzata in modo effettivo tramite la disponibilità delle applicazioni e la loro accessibilità. La ricchezza attraverso la *costumer-intimacy*, cioè la fidelizzazione del cliente, e la *transazionalità*, cioè il cliente interagisce con la parte fisica dell’azienda attraverso quella digitale in modo fluido (=ad esempio con risorse online chiare e aggiornate).

L’efficienza si traduce in *costo* e *tempo*.

I driver inoltre sono caratterizzati da una direzione di business.

L’estensione si traduce con una direzione *online* e *on-time*, cioè il SI deve essere accessibile online e più in real time possibile. La ricchezza si traduce con *l’enanched-CRM*, cioè gestione avanzata della relazione con il cliente, cioè avere una relazione di alto libello e stretta con il cliente, e con l’approccio *bricks-and-clicks*, che significa mischiare interazioni non-tecnologiche e interazioni tecnologiche con il cliente.

L’efficienza si traduce in business completamente *automatizzato* e business *compresso temporalmente*, cioè si concentra sull’utilizzare meno risorse umane e meno tempo possibili.

Realizzazione concreta della vista

Per realizzare concretamente la vista di business, data la descrizione di una situazione, gli step sono:

1. Identificare i partecipanti e il tipo di interazione (B2B, B2C, …?)
2. Identificare il tipo di oggetti scambiati
3. Identificare l’orizzonte temporale (che tipo di relazioni l’azienda stringe con i clienti? Durature, temporanee, …? Quindi stabili, dinamiche…?)
4. Identificare gli obiettivi da raggiungere con la realizzazione/evoluzione del SI (= i **driver**)
5. Specificare quali delle direzioni di business e dei driver effettivi sono applicabili alla situazione descritta

**VISTA ORGANIZZATIVA**

La vista organizzativa è articolata su tre livelli:

* Strutture dell’organizzazione e-business: identificazione dei partecipanti (parti) che interagiscono con l’organizzazione
* Funzioni di e-business: descrive le funzioni organizzative
* Processi di e-business: identifica i processi come sequenza di interazioni fra i partecipanti.

Parti

Le parti che interagiscono a livello organizzativo possono assumere i seguenti ruoli:

* *Consumatore*, è la parte che richiede un oggetto e-business;
* *Fornitore*, è la parte che fornisce un oggetto e-business;
* *Intermediario*, è la parte che ha un ruolo ausiliario nel trasferimento di un oggetto e-business dal fornitore al consumatore.

Gli intermediari possono essere:

* *Broker* che aiuta le parti a trovarsi sul mercato, cioè mette in contatto consumatori e fornitori
* *Intermediario* *finanziario* che gestisce i pagamenti
* *Intermediario* *di* *trasporto* che gestisce trasporto fisico di oggetti

Un altro elemento distintivo degli intermediari è il canale di comunicazione utilizzato per mettere in contatto consumatore e fornitore. Se diversi canali forniscono la stessa funzione di business, si parla di progetto multi-canale.

Funzioni

Le varie *parti* agiscono attraverso *funzioni di business* di due tipi diversi:

* L’aspetto *back-end*, che consiste nelle funzionalità interne all’azienda, cioè non esposte all’esterno;
* L’aspetto *front-end*, che consiste nelle funzionalità esterne all’azienda, che comunicano col mondo esterno.

L’aspetto front-end deve fornire varie funzionalità (pubblicizzare prodotti, negoziare prezzi, …). Le varie funzionalità sono assegnate a blocchi organizzativi diversi, detti moduli organizzativi.

Anche l’aspetto back-end può essere suddiviso in moduli, ma questo riguarda l’aspetto di Architettura (A) più che quello organizzativo.

Se l’azienda usa canali di comunicazione multipli, è utile avere un modulo diverso per ogni canale.

Alcune organizzazioni hanno un modulo intermedio, il mid-office. Il mid-office collega le funzionalità back-end a quelle front-end in modo flessibile. In pratica, il mid-office svolge una funzione di integrazione fra i due tipi di funzionalità.

Le varie parti interagiscono fra loro attraverso processi inter-organizzazione. Nel progetto di un SI, per mostrare che un processo è inter-organizzazione basta specificare chi svolge quale azione all’interno del processo, evidenziando che nello scenario agiscono più parti per quel processo. Un processo inter-organizzazione può avere un flusso di controllo unilaterale se solo una delle due parti prende le decisioni, e l’altra esegue piccole azioni su ordine della prima; bilaterale se il processo è diviso in due porzioni e ognuna delle due parti è responsabile solo di una; multilaterale se siamo nel caso bilaterale ma con più di due parti coinvolte.

Processi

Un processo di business è un insieme di attività eseguite in un certo ordine per raggiungere un determinato obiettivo di business.

Le attività (cioè i vari passi) di un processo sono eseguite da attori. Ogni attore ha un certo ruolo: il concetto di ruolo descrive le capacità che un attore deve avere per svolgere una certa attività. Selezionare un attor per ricoprire un certo ruolo viene detto *risoluzione di ruolo*.

Per descrivere un processo, si usa un modello di processo, cioè una descrizione che specifica l’ordine dei vari passi del processo. I passi possono essere attività elementari (cioè che non sono a loro volta processi) oppure sotto-processi, cioè processi a livello più basso.

L’ordinamento dei passi nel tempo è detto flusso di controllo. Una singola esecuzione di un processo è detto istanza di modello.

Il modello di un processo deve descrivere la sequenza di attività che lo compongono ed eventuali relazioni con altri processi.

Una notazione per modellare i processi è la *Business Process Modeling Notation.*

Ci sono tre elementi principali in un processo:

1. *Attività*, sono le unità di lavoro del processo, i vari passi. In BPMN è rappresentata come rettangolo con all’interno il nome dell’attività.
2. *Gateway*: sono i nodi di scelta del processo, cioè i momenti in cui in base a una condizione si intraprende un’attività piuttosto che un’altra. In BPMN sono rappresentati come rombi.
3. *Eventi*: sono informazioni relative all’ambiente in cui viene eseguito il processo. In BPMN sono rappresentati come cerchi.

I Gateway possono essere di tipo *XOR* (rappresentati da una X nel rombo) oppure *AND* (rappresentati da un + nel rombo). Gli XOR fanno scegliere un solo flusso fra due in base a una condizione. Gli AND permettono di seguire più flussi in parallelo.

I gateway possono essere usati anche per ricongiungere flussi, non solo per sceglierli o dividerli. Quando è usato per ricongiungere i flussi, lo XOR aspetta che uno dei due flussi da aspettare abbia terminato e poi fa procedere. L’AND invece aspetta che abbiano terminato tutti.

Gli eventi possono essere eventi *start* (fanno iniziare il processo, rappresentati con cerchio singolo), eventi *end* (fanno terminare il processo, rappresentati con cerchio singolo) o eventi *intermedi* (rappresentati con cerchio doppio). Alcuni eventi sono i timer (indicano eventi di natura temporale), eventi messaggio (indicano scambi di informazioni), …

Gli eventi si distinguono anche in base all’origine: se sono generati dal processo sono eventi *throw*, se sono “ricevuti” dal processo sono eventi *catch*.

I processi si dividono in due categorie:

* Processi statici
* Processi dinamici

*Processo statico*: un processo è statico se la sua specifica è completamente definita a priori, senza considerare nessuna interazione particolare, nessun ordine particolare da consumatore a fornitore. Le parti coinvolte sono gli unici elementi variabili del processo; tutte le attività e il flusso di controllo rimangono immutati in ogni istanza del processo.

*Processo dinamico*: un processo si dice dinamico se la sua specifica completa è determinata appena prima di eseguire un’istanza o durante l’esecuzione stessa. Per indicare che la specifica non è completa, si lasciano dei “blank” nella specifica, che vengono riempiti durante l’esecuzione dell’istanza.

Prima di procedere alla modellazione BPMN dei processi, occorre identificare i vari processi dell’organizzazione.

Un metodo semplice per fare questo è basarsi sulla similarità per settore: si guarda una lista di processi tipici, diffusi, per organizzazioni del settore dell’organizzazione di cui ci stiamo occupando, e si identificano quelli che compaiono nell’organizzazione in esame. Questo approccio viene detto *check-list*. È un metodo poco efficiente, perché funziona bene solo per settori in cui i processi aziendali sono molto simili.

Un altro metodo per identificare i processi è usare la Catena del Valore di Porter: si prendono come processi le categorie stabilite dalla Catena di Porter, e si associa ogni processo alle funzioni di business (identificate nell’analisi front-end vs. back end) che sono legate ad esso. In pratica, si trovano tutte le funzioni back-end e front-end; si sceglie se occuparsi prima di quelle front-end o back-end; si prendono tutte le funzioni della categoria scelta, si prendono tutti i processi della catena di Porter (ad esempio procurement, marketing, …) e si decide a quale di questi appartenga ogni funzione; poi si fa la stessa cosa per l’altra categoria di funzioni (quindi se per prime ho divido le funzioni di back-end, ora classificherò quelle di front-end). Lo schema così ottenuto identifica i processi e associa ad essi le funzioni di business dell’organizzazione che li riguardano.

Un terzo metodo è l’approccio analitico.

L’approccio analitico è un metodo top-down che agisce così: si identificano per prima cosa i destinatari del processo, cioè le altre parti con cui la nostra organizzazione interagisce; si identifica l’output a loro fornito dal processo; si ricostruiscono le attività che hanno prodotto quell’output; si ricostruiscono i ruoli coinvolti in quell’attività; si ricostruiscono infine gli input (cioè le materie impiegate e le informazioni fornite ai vari ruoli) dati per giungere a quell’output.

I business driver dell’aspetto O possono essere operazionalizzati (=cioè implementati in modo concreto) così:

* Il driver *reach* è operazionalizzato sotto forma dei canali di comunicazione usati dalle varie parti per interagire.
* Il driver *richness* dipende dai blocchi funzionali del front-end: per aumentare la richness bisogna espandere e aumentare i moduli.

**ASPETTI ARCHITETTURALI**

Un’architettura funzionale di un SI specifica la struttura del SI dal punto di vista delle componenti software e delle interfacce attraverso cui esse comunicano.

Si possono descrivere le architetture dei SI su vari livelli. Per distinguere un livello dall’altro, si usano due dimensioni: aggregazione e astrazione.

Livelli di aggregazione

La dimensione aggregazione riguarda il grado di dettaglio e la dimensione della struttura in esame. Diminuendo il livello di aggregazione, si fa uno “zoom” dentro l’architettura, analizzando strutture più piccolo con maggiore dettaglio. Aumentando il livello di aggregazione, si fa uno “zoom fuori” dall’architettura, analizzando strutture più grandi in minor dettaglio.

Si possono distinguere tre livelli di aggregazione per le strutture (dal più basso al più alto):

* *Architetture Market level*: descrivono la struttura del SI al livello in cui le varie parti si inseriscono nel mercato.
* *Architetture Party level*: descrivono la struttura a livello di singole organizzazioni.
* *Architetture System level*: descrivono le strutture dei sottosistemi di una delle parti.

Architetture Market Level

Descrive l’architettura di un intero SI e si concentra sui messaggi scambiati fra parti. Definisce una struttura a livello inter-organizzativo (cioè descrive i SI di tutti i partecipanti, non solo di uno) in termini di componenti software che supportano specifiche funzioni delle parti coinvolte e di interazioni fra queste componenti. La parte principale di questo livello è quella che riguarda le interfacce che permettono le interazioni fra parti.

Il Market Level riguarda dunque i sistemi front-end delle varie parti e il modo in cui essi comunicano. Va notato che non c’è per forza una corrispondenza 1:1 fra sistemi front-end e funzioni front-end: una funzione può richiedere vari sistemi o un sistema può implementare più funzioni.

Architetture Party Level

Un’architettura party level descrive i SI di ogni singolo partecipante, quindi è di tipo intra-organizzativo. Si occupa principalmente di sistemi back-end.

I componenti delle architetture party level sono:

* Componenti generici, cioè i principali componenti funzionali
* Database e DBMS, denotano i componenti che si occupano della gestione e memorizzazione e accesso dei dati
* Connettori, sono i componenti che collegano tutti gli altri

L’architettura party level è un sottoinsieme dell’Enterprise Architecture, adattata a uno specifico scenario. Spesso i sistemi vengono parzialmente riutilizzati in vari scenari, quindi le architetture party level si sovrappongono per i vari scenari nell’EA, cioè hanno alcune funzionalità in comune.

Architetture system level

L’architettura system level descrive la struttura di uno specifico sistema informativo di un partecipante dello scenario, in termini di componenti software e interfacce. Si tratta di un’architettura intra-organizzativa, ma può essere sia back-end sia front-end a seconda dello scopo del SI descritto.

Livelli di astrazione

Aumentando il livello di astrazione, i componenti sono descritti in modo molto generico, specificando ad esempio solo le funzionalità. Diminuendo il livello di astrazione, le componenti sono descritte in modo molto concreto, con precisione, determinando la funzionalità con esattezza e specificando dati come il nome del software, la versione del software, …

Dal punto di vista dell’astrazione, possiamo dividere le architetture su tre livelli (da quello con minore astrazione a quello con maggiore astrazione):

* *Architettura istanza*: è una specifica istanza di uno specifico SI; è molto specifica nelle sue funzionalità, non è pensata per essere riusata per contesti diverso da quello per cui è nata;
* *Architettura standard*: è un’architettura definita come standard per una classe di SI in un contesto (cioè per una determinata organizzazione), usata come base per definite un’architettura istanza;
* *Architettura di riferimento*: è un’architettura definita come standard per una classe di SI per varie organizzazioni, non per una sola. Può essere usata per definire architetture standard, architetture istanza o altre architetture di riferimento.