

CBAM – Meccanismo di Persistenza dei Dati



Gian Marco Falcone - 0300251

Indice

- Introduzione: CBAM
- Step 1: Definizione della decisione
- Step 1: Le alternative
- Step 2: Gli attributi di qualità
- Step 3: Valutazione delle alternative
- Step 4: Calcolo del rischio
- Step 5: Calcolo dei benefici
- Step 6: Valutazione dei costi
- Step 7: Calcolo della desiderabilità
- Step 8: Classifica

Introduzione: CBAM

CBAM è l'acronimo di Cost-Benefit Analysis Method.

Si tratta di una tecnica decisionale che le aziende, e non solo, utilizzano per analizzare quali decisioni prendere e a quali rinunciare.

L'analisi costi-benefici somma i potenziali benefici attesi da un'alternativa (o scelta) e quindi sottrae i costi totali associati al suo utilizzo.

Spesso i costi vengono calcolati in dollari (\$).

L'utilità di condurre un'analisi di questo tipo, prima di intraprendere un nuovo progetto o prima di effettuare una certa scelta, è data dalla possibilità di valutare i potenziali costi e ricavi che si potrebbero generare dal progetto o dalla scelta in questione.

Step 1: Definizione della decisione

Si vuole applicare la tecnica CBAM per decidere quale meccanismo di persistenza dei dati sia preferibile utilizzare nel progetto software sotto sviluppo.

I fattori da valutare sono:

- La quantità di dati da gestire non è molto grande.
- È necessario considerare che più utenti possano accedere in contemporanea agli stessi dati.
- I dati devono rimanere sempre coerenti e aggiornati.
- Non è ammessa la presenza di guasti che inficiano sul corretto utilizzo del software.
- Sono richieste delle performance adeguate.

Step 1: Le alternative

In base alle considerazioni fatte in precedenza, le alternative tra cui scegliere sono:

- File system
- Database MySQL
- Database NoSQL open-source
- Database SQL non open-source

Step 2: Gli attributi di qualità

1. Performance	6
2. Sicurezza	10
3. Competenza nell'utilizzo	17
4. Semplicità di configurazione	13
5. Affidabilità	11
6. Disponibilità	7
7. Tolleranza ai guasti	9
8. Facilità di integrazione	12
9. Coerenza tra i dati	15

Step 3: Valutazione delle alternative (1)

File system

1. Performance	0.4
2. Sicurezza	0.2
3. Competenza nell'utilizzo	1
4. Semplicità di configurazione	1
5. Affidabilità	-0.2
6. Disponibilità	0.5
7. Tolleranza ai guasti	-0.4
8. Facilità di integrazione	1
9. Coerenza tra i dati	-0.1

Step 3: Valutazione delle alternative (2)

Database MySQL

1.	Performance	0.5
2.	Sicurezza	0.4
3.	Competenza nell'utilizzo	0.9
4.	Semplicità di configurazione	0.2
5.	Affidabilità	0.6
6.	Disponibilità	0.4
7.	Tolleranza ai guasti	0.3
8.	Facilità di integrazione	0.6
9.	Coerenza tra i dati	1

Step 3: Valutazione delle alternative (3)

Database NoSQL open-source

1. Performance	0.7
2. Sicurezza	0.4
3. Competenza nell'utilizzo	-0.1
4. Semplicità di configurazione	-0.3
5. Affidabilità	0.6
6. Disponibilità	0.5
7. Tolleranza ai guasti	0.6
8. Facilità di integrazione	0.3
9. Coerenza tra i dati	0.4

Step 3: Valutazione delle alternative (4)

Database SQL non open-source

1. Performance	0.6
2. Sicurezza	0.7
3. Competenza nell'utilizzo	0.8
4. Semplicità di configurazione	0.5
5. Affidabilità	0.8
6. Disponibilità	0.9
7. Tolleranza ai guasti	0.7
8. Facilità di integrazione	0.8
9. Coerenza tra i dati	1

Step 4: Calcolo del rischio

- File system 0.3
- Database MySQL 0
- Database NoSQL open-source 0.4
- Database SQL non open-source 0

Avendo già familiarità nell'utilizzo di database relazionali, non c'è alcun rischio nell'adottare questa soluzione nel progetto.

Il rischio sul file system è stato calcolato in base alla possibilità che la mole di dati da gestire cresca notevolmente nel tempo e renda l'utilizzo di tale alternativa non più adeguato.

Riguardo il database NoSQL, il rischio è riferito al fatto che non si è mai fatto uso di tale strumento; l'alternativa potrebbe portare con sé diverse problematiche non considerate.

Step 5: Calcolo dei benefici

In base ai valori assegnati in precedenza risulta che:

- Benefit(File system) $= 42.6 * (1 - 0.3) = 29.8$
- Benefit(Database MySQL) $= 59.2 * (1 - 0) = 59.2$
- Benefit(Database NoSQL open-source) $= 27.7 * (1 - 0.4) = 16.6$
- Benefit(Database SQL no open-source) $= 76.7 * (1 - 0) = 76.7$

Step 6: Valutazione dei costi

Per semplicità di ragionamento associamo ai costi valori indicativi.

Tutte le alternative hanno bisogno di manutenzione nel tempo. Supponiamo che la necessità di controllarne periodicamente il corretto funzionamento abbia un costo pari a 10.

Le alternative File System, Database MySQL e Database NoSQL sono tutte open-source: per queste ad esse non viene associato nessun costo ulteriore.

Il Database SQL no open-source richiede invece l'acquisto di una licenza mensile per essere utilizzato all'interno del progetto: assumiamo che tale costo sia pari a 5.

Quindi abbiamo:

$\text{Costo}(\text{File System}) = \text{Costo}(\text{Database MySQL}) = \text{Costo}(\text{Database NoSQL open-source}) = 10$

$\text{Costo}(\text{Database SQL no open-source}) = 15$

Step 7: calcolo della desiderabilità

Calcoliamo la desiderabilità di ogni alternativa secondo la formula:

➤ $\text{Desiderabilità}(\textit{alternativa}) = \text{Beneficio}(\textit{alternativa}) / \text{Costo}(\textit{alternativa})$

Dai risultati precedenti otteniamo che:

- Desiderabilità(File system) = $29.8 / 10 = 2.98$
- Desiderabilità(Database MySQL) = $59.2 / 10 = 5.92$
- Desiderabilità(Database NoSQL open-source) = $16.6 / 10 = 1.66$
- Desiderabilità(Database SQL no open-source) = $76.7 / 15 = 5.11$

Step 8: Classifica

Possiamo così ordinare le nostre alternative, in ordine decrescente, in base alla desiderabilità:

1. Database MySQL
2. Database SQL no open-source
3. File system
4. Database NoSQL open-source