



Deliverable 3

CBAM (COST-BENEFIT ANALYSIS METHOD)

MARTINA SALVATI 0292307

Contenuto delle diapositive

Introduzione

- Case Study : Graphs

Progettazione CBAM

1. Definire la decisione



2. Definizione degli attributi di qualità



3. Classificazione di ogni alternativa



4. Calcolo del rischio



5. Calcolo del punteggio di Benefit



6. Valutazione costo



7. Calcolo dell'auspicabilità di ogni alternativa



8. Classificazione dell'alternativa

Conclusione

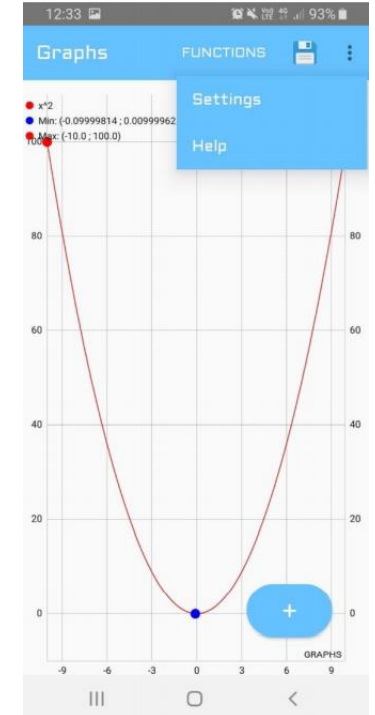
- Considerazioni finali

Case Study : Graphs

Graphs è il progetto che è stato eseguito durante lo svolgimento della laurea triennale nel corso di Mobile programming da Martina Salvati, Davide Salerno e Paolo Melissari

[See details about project](#)

- ❖ Graphs è un applicazione android che permette di acquisire dall'utente una funzione $f(x)$ ed eseguirne il plot nell'intervallo $[a,b]$, gestendo i vari errori legati al dominio della funzione
- ❖ Inoltre permette la *memorizzazione* delle funzioni inserite dall'utente (funzionalità 'memo')
- ❖ L'applicazione graphs viene realizzata in *AndroidStudio* e scritta in Java
- ❖ Nel corso di sviluppo dell'applicazione ci si è chiesto quale fosse il metodo migliore per lo sviluppo della persistenza dell'applicazione



Progettazione : definire la decisione

Room vs SQLite

Quale metodo di persistenza utilizzare per il salvataggio dei grafici?

- ❖ Serve individuare una metodologia per effettuare il salvataggio delle funzioni inserite dall'utente.
- ❖ Due alternative possibili

Room

- E' una libreria ORM – Object Relation Mapping
- La libreria crea una cache dei dati dell'app sul dispositivo che esegue l'app
- Necessita configurazione
- Offre una astrazione del linguaggio SQL
- Facile da utilizzare

SQLite

- SQ-Lite è RDBMS
- Per utilizzare la libreria bisogna utilizzare il linguaggio SQLite
- Non necessita di configurazione in quanto libreria nativa
- Complesso utilizzo
- Supporto continuo dagli sviluppatori

Progettazione: definizione degli attributi di qualità

Non tutti gli attributi di qualità sono ugualmente importanti per uno stakeholder; alcuni sono più desiderabili di altri.

In contesto CBAM si utilizza il meccanismo a pesi diretti per descrivere la misura degli attributi di qualità.

Gli attributi di qualità individuati sono :

Availability

- La Availability si riferisce alla capacità del servizio di essere disponibile e accessibile
- QAScore : 7,5

Security

- La sicurezza si riferisce al livello di sicurezza offerto dalla libreria per l'accesso ai dati
- QAScore : 20

Performance

- La performance si riferisce ai tempi di latenza delle operazioni CRUD
- QAScore : 30

Usability

- L'usability si riferisce alla qualità del software di essere user-friendly, focalizzandosi sull'esperienza utente.
- QAScore : 20

Maintainability

- La Maintainability si riferisce alla qualità del software di supportare modifiche
- QAScore : 7,5

Modifiability

- La modifiability si riferisce alla misura in cui l'architettura del database può essere modificata in modo efficiente con compromessi e difetti minimi
- QAScore : 15

Progettazione: Classificazione di ogni alternativa

L'obiettivo del task è classificare ogni alternativa per ogni attributo di qualità con un punteggio nell'intervallo [-1, +1].

	Quality	Room	SQ-Lite
Availability		0,9	0,7
Security		0,8	0,8
Performance		0,8	0,3
Usability		0,9	0,7
Maintainability		0,9	1
Modifiability		1	0,4

Progettazione : Calcolo del rischio

Calcolare il rischio per ogni alternativa con un punteggio nell'intervallo $[0, 1]$. Più alto è il punteggio maggiore è il rischio, peggiore è l'alternativa.

LIBRERIA	RISCHIO
Room	0,3
SQ-Lite	0,1

- ❖ Il rischio di Room è il fatto che non sia una libreria nativa di AndroidStudio
- ❖ Il rischio di SQ-Lite, minimo, riguarda le difficoltà dell'utilizzo

Progettazione: Calcolo del punteggio di Benefit

Calcolo del punteggio di benefit
per ogni alternativa

Quality	Room	SQ-Lite	QAScore
Availability	6,75	5,25	7,5
Security	16	16	20
Performance	24	9	30
Usability	18	14	20
Maintainability	6,75	7,5	7,5
	15	6	15
Rischio	0,3	0,1	
Benefit	60,55	51,975	

$$Benefit(AS_i) = (\sum_j (Cont_{i,j} * QAscore_j)) * |Risk_i - 1|$$

Progettazione: Valutazione Costo

Valutazione del costo di ogni alternativa in \$.

- ❖ Entrambe le librerie sono open-source, quindi il costo viene impostato pari a 1
- ❖ $\text{Cost}(\text{Room})=1$
- ❖ $\text{Cost}(\text{SQLite})=1$

Desirability	Room	SQ-lite
$Desirability(Asi) = \frac{Benefit(Asi)}{Cost(Asi)}$	$\frac{Benefit}{Cost} = \frac{60,55}{1} = 60,55$	$\frac{Benefit}{Cost} = \frac{51,975}{1} = 51,975$

PROGETTAZIONE : CALCOLO DELLA DESIDERABILITÀ DI OGNI ALTERNATIVA

Progettazione: Classificazione dell'alternativa

Classificare l'alternativa in base
a desiderabilità in ordine
decrescente

1.Room

2.SQLite

Conclusioni

- ❖ CBAM è un metodo che fa sì che gli stakeholders quantifichino i vantaggi, nonché i costi, le dipendenze e le implicazioni di pianificazione delle decisioni architettoniche
- ❖ Il metodo ha permesso una facilitazione nella scelta progettuale riguardante la libreria che permette la consistenza delle funzioni dell'applicazione

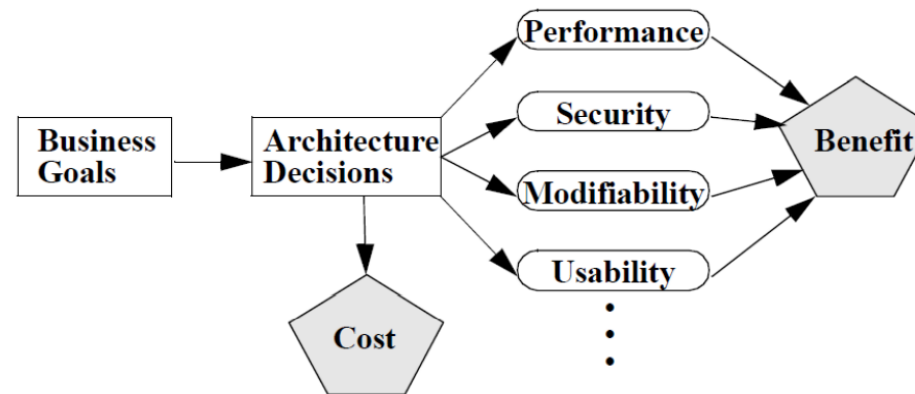


Figure 1. Context for the CBAM