**Ingegneria ed Evoluzione del Software**

Coordinatore:**Prof.Andrea De Lucia**

Studente:**Pasquale Lombardi**

Progetto:Agenzia immobiliare

Documento:impact analysis

Il documento di impact Analysis si prefigge di analizzare i possibili effetti che si possono riscontrare durante la fase di manutenzione del software di una applicazione.

L’insieme degli artefatti(Task)che non rispettano le l’IEE dell’applicazione sono definiti False candidati Positivi mentre l’insieme degli artefatti che rispettano l’IEE dell’ applicazione vengono definiti False candidati Negativi.

Pertanto l’ Impact analysis ha esito favorevole solo quando viene individuato un insieme di falsi Negativi,ed solo dopo di ciò è possibile effettuare la manutenzione dell’Applicazione,ciò avviene rideterminando gli scenari dell’ applicazione,i casi d’uso,i rischi ed i costi che tale operazione comporta.

Tale fase viene denominata re-ingegnirizazione dell’applicazione, in tale fase vengono ereditati le dipendenze funzionali dell’artefatto in oggetto e mappate con l’artefatto da includere nella applicazione e solo dopo aver concluso tale operazione viene effettuata l’operazione di testing .

Durante tale fase viene testata la effettiva vulnerabilità del software ovvero la capacità dello stesso di rispondere ai requisiti funzionali per il quale esso è stato progettato ed solo dopo aver riscontrato la effettiva capacità esso viene validato.

Nel esempio in oggetto viene preso in considerazione il prodotto software, agenzia immobiliare.

In tale prodotto software si intende aggiungere una nuova funzionalità: Acquista immobile/vendita immobile per effettuare tale operazione di manutenzione ridisegnato l’intera architettura del prodotto software ovvero vengono mappati tutti task del prodotto software con il nuovo task in modo da ottenere la corretta funzionalità dello stesso.

A tale fine viene effettuato lo stess test dell’intero prodotto ovvero viene misurato la grandezza dello stesso ed se è in grado di sopportare il carico di input per il quale esso è stato progettato,le iterrogazioni che può sopportare,il numero di output che può dare.

Introduzione

* 1. Scopo del documento
  2. Architettura Software proposta
  3. Obbiettivi dell’impact analysis
  4. Decomposizione in sottosistemi
  5. Mapping Hardware/software
  6. Gestione dei dati persistenti
  7. Controllo degli accessi
  8. Operazioni effettuabili dall’utente
  9. Operazioni effettuabili dall’amministratore
  10. Flusso di controllo
  11. Condizioni limite

1.introduzione

* 1. scopo del documento e del sistema.

Il documento di impact analysis avrà il compito di descrivere le qualità del sistema,con particolare riferimento all’architettura software e quindi alla decomposizione del sistema in sottosistemi.

* 1. Architettura Software proposta.

Il sistema è progettato per immagazzinare una notevole quantità di dati essendo dipendente dal numero degli articoli presenti nella agenzia.

1.3 Gli obbiettivi di design identificati nel prodotto software prima della manutenzione rispecchiano cinque tipologie di ategorie distinte ed le stesse devono essere rispettate anche dopo la manutenzione:

* Performace criteria
* Dependability criteria
* Maintenance criteria
* Cost criteria
* End User criteria

Performance Criteria

Tempo di risposta

Il tempo di risposta è stato preventivato in un tempo nullo o quasi, quantificabile in un secondo,il tempo impiegato per la rilevazione degli articoli presenti nel sistema, è di cinque secondi

Memoria

L’occupazione principale di memoria è ad opera del DBMS che la usa per contenere l’insieme dei dati degli articoli delle entità del nostro sistema. In secondo luogo, la gestione della memoria è delegata alla JVM, che periodicamente effettua la garbage collection per liberare aree di memoria non più utilizzate.

*Dependability Criteria*

*Robustezza*

Il sistema sarà in grado di gestire eventuali input errati da parte degli utenti senza che questi modifichino le sue funzioni.

*Attendibilità*

Il sistema rispetterà le specifiche fornite dal cliente specificate nel documento di analisi dei

requisiti.

*Tolleranza ai fault*

Il sistema sarà in grado di gestire errori sia dovuti all’interazione con attori esterni sia ad eventuali problemi interni. In entrambi i casi verrà visualizzato un messaggio specifico in base all’eccezione generata.

*Disponibilità*

Il sistema sarà disponibile durante l’attività lavorativa degli utenti.

*Sicurezza*

La sicurezza verrà garantita attraverso sistemi d’autenticazione con login e password e le

funzionalità gestionali del sistema verranno messe a disposizione solo agli utenti aventi privilegi necessari.

*Maintenance Criteria*

*Estendibilità*

Risulterà abbastanza semplice estendere classi esistenti e aggiungere nuove funzionalità grazie alla flessibilità del linguaggio usato.

*Adattabilità*

Il sistema potrà adattarsi facilmente a nuovi strumenti elettronici di rilevazione, ma risulterà

complesso adattarlo a nuovi domini di applicazione.

*Portabilità*

Il sistema sarà indipendente dal tipo di piattaforma utilizzata.

*Leggibilità*

Il codice del progetto deve essere correttamente documentato e strutturato, in questo modo saranno possibili veloci modifiche o estensioni del sistema.

*Cost Criteria*

Il nostro software è stato adeguatamente progettato per avere un costo contenuto.

*End User Criteria*

*Utilità*

Il sistema utilizzerà delle interfacce grafiche intuitive e di facile utilizzo, per garantire un aspetto gradevole all'utilizzatore del programma; l'utente infatti, troverà dinnanzi a sé una struttura grafica semplice, completa ed essenziale.

*Usabilità*

Il sistema deve risultare semplice da utilizzare anche per utenti non esperti; le interfacce che si susseguono infatti, avranno un aspetto simile fra loro, e le operazioni saranno facilmente eseguibili grazie ad interfacce intuitive che guideranno il cliente passo dopo passo durante lo svolgimento dell’operazione

**1.4 Decomposizione in sottosistemi**

Il sistema segue lo stile architetturale three-tier in versione Client/Server. L'organizzazione delsistema secondo il modello three-tier prevede la suddivisione dello stesso in tre strati:

*Interface Layer:*

Include tutte le interfacce grafiche e in generale i boundary objects, come le form e i link, con cuiinteragisce l'utente.

*Application Layer:*

Include tutti gli oggetti relativi al controllo e alle entità del sistema.

*Storage Layer:*

Effettua la memorizzazione, il recupero e l'interrogazione degli oggetti persistenti.



*Architettura three-tier*

Nello stile architetturale Client/Server, un sottosistema, il Server, provvede a fornire servizi alle istanze di altri sottosistemi chiamati Client, che sono responsabili dell'interazione tra utenti e sistema. Questi ultimi dunque sono responsabili della raccolta degli input immessi dagli utenti e delle inizializzazioni delle transazioni del database quando tutti i dati necessari sono stati raccolti.

Interfaccia Agente immobiliare Interfaccia Utente

Visualizza articoli Gestione agenzia immobiliare

Visualizza immobile

Acquista immobile

Inserisci immobile

Visualizza immobile

Cancella immobile

Agenzia immobiliare

1)Rettangolo Presentation Layer

2)Rettangolo Application Layer

3)Rettangolo Storage Layer

Nello stile architetturale del sofware agenzia Immobiliare il Presentation layer è caratterizzato dalle interfacce grafiche degli utenti, ovvero dall’interfaccia dell utente e dell’amministratore.

La logica applicativa contiene tre sottosistemi: la gestione del DBC, che è associata all’interfaccia dell’agente immobilare; la visualizzazione dei prodotti che è associata sia all’interfaccia dello utente che all’interfaccia dell’agente.

Lo Storage layer invece conterrà la base di dati dove verranno resi persistenti i dati, e comunicherà con l’application layer attraverso il framework JDBC.

Il compito dell’impact analysis è quello di decomporre il sistema in sottosistemi che mantengano tra di loro un giusto equilibrio tra coesione e accoppiamento per permettere cosi ai team di lavorare sui sottosistemi in modo individuale, con un minimo overhead di comunicazione ed successivamente integrare la parte restante del prodotto software con il task,che rappresenta la nuova funzionalità del prodotto software.acquista immobile.

A tal proposito, si è deciso di suddividere il sistema in tre macroaree, come evidenziato anche nella scelta dello stile architetturale.

Il primo sottosistema evidenziato è “Gestione interfacce Utente e Amministratore”.

Si è deciso di raggrupparli in un unico sottosistema in quanto tra di loro c’è una forte coesione, dovuta alle caratteristiche comuni degli utenti.

Il secondo sottosistema, “Gestione immobili presenti nel Dbc dell’agenzia immobiliare” vengono effettuate da parte dell’agente.

.

**1.5 Mapping Hardware/Software**

Dispositivo di rilevazione

Client Application

Server

Mysql

Dispositivo di Rilevazione

Local Mysql

Local Mysql

Client Application

Il sistema,piattaforma per la gestione dell’agenzia immobiliare, come detto, si basa su un’architettura Client-Server. Il Deployment diagram proposto descrive il sistema in termini di risorse hardware e di relazioni tra esse. Come si evince dal diagramma, il server centrale metterà a disposizione dei client la funzionalità di sincronizzazione,che permetterà ai database locali di trasmettere le nuove informazioni acquisite al DBMS centrale e,allo stesso tempo, di percepire eventuali aggiornamenti. Il modello di comunicazione è basato su HTTP.

Tutti i client avranno a loro disposizione, oltre ad un database locale, un lettore badge che

consentirà la rilevazione delle presenze. Il modello di comunicazione con i dispositivi sarà interno al nostro software.

**1.6 Gestione dai dati persistenti**

Analizzando le caratteristiche del nostro sistema, abbiamo individuato e raggruppato nel seguente

elenco le entità che dovranno essere rese persistenti:

- Amministratore;

- Utente;

-Prodotti(immobili);

Abbiamo optato per la realizzazione di un Database relazionale, in quanto il sistema necessita di una tecnologia matura, che supporta letture e scritture multiple, query complesse e un ottimo livello di sicurezza. Inoltre, è possibile gestire le relazioni che intercorrono tra le varie entità con relativa facilità.

**1.7 Controllo degli accessi**

Nel nostro sistema, attori diversi hanno accesso a funzionalità e informazioni diverse. Il controllo degli accessi mediante autenticazione risulta abbastanza efficace nella gestione della piattaforma dell’agenzia immobiliare.

Più in dettaglio, andiamo a specificare le funzionalità utilizzabili da ciascun utente.

**1.8 operazioni effettuabili dall’utente prima della manutenzione:**

-Accedi al sistema

-Visualizza immobile

**1.8.1 Operazioni effettuabili dall’utente dopo la manutenzione:**

Lutente può accedere alle funzionalità:

- Accedi al sistema;

- Visualizza immobili;

-Acquista immobili;

**1.9 Operazioni effettuabili dall’Agente immobiliare**

L’Agente immobilare può accedere alle funzionalità:

-Inserisci immobile;

- Ricerca immobile;

-Cancellazione immobile;

**1.10 Flusso di controllo globale**

Nella parte Client, il paradigma utilizzato è l’event-driven, perché le varie azioni da eseguire sono dettate da eventi collegati ai vari oggetti che compongono la form.

Problemi legati alla concorrenza possono verificarsi solo in caso di sincronizzazione contemporanea da parte di due client. Nella piattaforma “Agenzia Immobiliare”, al primo client che richiederà la sincronizzazione dei dati sarà consentita l'operazione, mentre i client seguenti saranno messi in lista d'attesa.

**1.11 Condizioni limite**

Le condizioni limite che possono verificarsi nella piattaforma “Agenzia immobiliare” riguardano l’assenza di connessione al server centrale: in questo caso può verificarsi un solo problema, ovvero l’impossibilità di sincronizzazione dei dati con il server centrale.

Il problema però non dipende dal software, e quindi non è gestibile dagli sviluppatori; Gli utenti che si troveranno di fronte a questo problema, potranno

solamente attendere che la connessione al server sia nuovamente disponibile.