Documentazione

UniMarket

Progetto di Ingegneria del Software

Di Davide Dell'Anno, Davide Carissoni, Francesca Corrente

Università degli Studi di Bergamo
Facoltà di Ingegneria Informatica
2024-2025

Indice

1 Introduzione

2 Modelli di Processo

Lo sviluppo del progetto seguirà un processo **RAD** (Rapid Application Development), con l'obiettivo di realizzare un'applicazione funzionante entro i tempi stabiliti dal *time box*. Verrà adottato il **modello di Kano** per classificare i requisiti del progetto in base alla loro capacità di soddisfare le preferenze del cliente.

3 Organizzazione del Progetto

Per questo progetto è stata creata una **Squadra SWAT**, ovvero un team agile e versatile, composto da membri con competenze complementari, che si adatta rapidamente ai cambiamenti del progetto. in questo modo è più facile rispondere prontamente alle esigenze del cliente e dell'ambiente di sviluppo, mantenendo un'elevata efficienza nella gestione delle attività.
La configurazione del team di sviluppo segue il paradigma **Agile** e si basa su un modello già collaudato. I ruoli sono stati assegnati come segue:

Scrum Master: X Ha il compito di coordinare le attività, pianificare i moduli per gli sprint e assicurarsi il rispetto delle scadenze.

Product Owner: Y Scelto per il suo elevato livello di interazione con il cliente e la profonda conoscenza del progetto.

Sviluppatori, Progettisti, Tester: X, Y, Z, W Vista la piccola grandezza del team, tutti i membri del progetto si occupano di tutte e tre le mansioni.

4 Standard, Linee Guida, Procedure

5 Attività di Gestione

Per garantire una gestione efficace delle attività e dello sviluppo del progetto, sono stati definiti i seguenti obiettivi:

Incontri settimanali: monitorare l'avanzamento del lavoro, affrontare eventuali problemi riscontrati, discutere strategie risolutive e valutare la direzione dello sviluppo.

Kanban board su GitHub: tracciare il progresso delle attività, organizzare il lavoro e assegnare incarichi ai membri del team.

Gestione del progetto con GitHub: tenere traccia delle modifiche, monitorare gli aggiornamenti e facilitare la collaborazione.

6 Rischi

I rischi a cui bisogna prestare più attenzione sono:

Compatibilità dei sistemi utilizzati: Il progetto viene sviluppato per essere utilizzato su desktop; pur usando maven, che consente il download automatico delle librerie richieste, potrebbero comunque esserci problemi di visualizzazione o calcolo su mobile o altri S.O. desktop.

Limitate quantità e tipologie di prodotto: UniMarket è indipendente da reali supermarket, non dispone di tante opzioni per ogni genere di prodotto.

Bugs: Ci potrebbero essere problemi generati dall'uso scorretto e imprevisto del programma da parte del cliente, è possibile che alcuni bug agli estremi dell'utilizzo improprio rimangano pur dopo i test.

7 Personale

8 Metodi e Tecniche

I metodi e le tecniche adottati per le diverse fasi dello sviluppo sono i seguenti:

Pianificazione dei requisiti: si basa sull'elicitazione e sull'analisi dei requisiti per definire in modo chiaro le funzionalità richieste dall'applicazione.

Progettazione dell'applicazione: prevede la definizione del funzionamento dell'applicazione, delle funzionalità incluse e degli strumenti software da utilizzare.

Programmazione: consiste nello sviluppo del codice sorgente dell'applicazione seguendo le specifiche progettuali.

Testing: include la scrittura e l'esecuzione di test, utilizzando strumenti appositi, per verificare il corretto funzionamento del codice e individuare eventuali anomalie.

Queste attività saranno supportate da un'adeguata gestione del controllo di versione e della configurazione dei componenti software, documentando ogni fase con precisione.

9 Garanzie di Qualità

Al fine di assicurare la qualità durante il processo di sviluppo e la realizzazione del software, verranno utilizzati i seguenti criteri:

Correttezza: Il software deve funzionare come previsto e soddisfare i requisiti specificati, senza errori o comportamenti imprevisti. La correttezza implica che il sistema esegua tutte le operazioni correttamente, restituendo i risultati attesi in ogni condizione di input.

Affidabilità: Il software deve essere stabile e operare in modo continuo e senza interruzioni. Un software affidabile deve resistere a guasti o malfunzionamenti, gestendo situazioni impreviste e mantenendo la sua funzionalità nel tempo, anche sotto carichi elevati o condizioni di stress.

Integrità: L'integrità riguarda la protezione dei dati e il mantenimento della coerenza e correttezza durante le operazioni. Un software con buona integrità assicura che i dati non vengano alterati o danneggiati involontariamente, né durante l'elaborazione né durante la memorizzazione.

Usabilità: Il software deve essere intuitivo e facile da usare per gli utenti finali. Una buona usabilità garantisce che gli utenti possano navigare nell'interfaccia senza difficoltà, riducendo il numero di errori operativi e migliorando l'esperienza complessiva.

Manutenibilità: Un software manutenibile è facilmente aggiornabile e modificabile nel tempo per adattarsi a nuove esigenze o correggere eventuali difetti. Ciò implica che il codice sia chiaro, ben strutturato e documentato, facilitando interventi futuri senza introdurre regressioni o problemi aggiuntivi.

10 Paccheti di lavoro

11 Risorse

Le risorse relative allo sviluppo software sono le seguenti:

Eclipse IDE: per lo sviluppo e la modifica del codice Java.

Maven: per la gestione e l'automazione dei progetti Java.

SQLite: per la creazione e gestione di un database embedded.

Vaadin: per la progettazione iniziale dell'interfaccia grafica dell'applicazione.

Strumenti aggiuntivi:

LaTeX: per la redazione della documentazione tecnica.

Papyrus: per la modellazione di diagrammi UML.

Le risorse hardware si limitano ai dispositivi personali dei membri del team.

12 Budget

Il budget del progetto, essendo privo di spese finanziarie, è relativo al tempo di lavoro necessario, ipotizzato intorno alle 60 ore per ciascun membro.

13 Cambiamenti

14 Consegna

La documentazione e l'applicazione finale saranno consegnate condividendo il progetto sul repository GitHub.