

ProjectPlan - CMMTickets



**UNIVERSITÀ
DEGLI STUDI
DI BERGAMO**

Caprinali Michele 1087210
Mazzoleni Gabriel 1086530
Mazzoleni Raphael 1086531

Indice:

1. [Introduzione](#)
2. [Modello di processo](#)
3. [Organizzazione del progetto](#)
4. [Standard, linee guida e procedure](#)
5. [Attività di gestione](#)
6. [Rischi](#)
7. [Personale](#)
8. [Metodi e tecniche](#)
9. [Garanzia di qualità](#)
10. [Work Packages](#)
11. [Risorse](#)
12. [Budget e programma](#)
13. [Cambiamenti](#)
14. [Consegna](#)

Introduzione

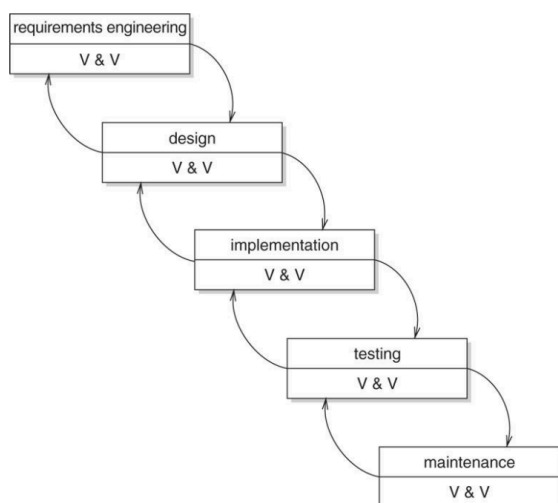
L'obiettivo del progetto è lo sviluppo di un software (versione desktop) chiamato CMMTickets, pensato per la gestione completa del processo di acquisto e vendita di biglietti per eventi di vario genere, come partite sportive, concerti e competizioni.

Gli utenti, che dovranno registrarsi al sistema, avranno la possibilità di selezionare gli eventi di loro interesse, acquistare un numero di biglietti definito in base ai limiti imposti per ciascun evento, e consultare i dettagli dei biglietti acquistati. Tali dettagli includeranno informazioni come data, luogo, orario, settore e posto. Gli eventi potranno disporre di posti numerati oppure posti unici non assegnati.

Il software prevede anche una sezione dedicata agli amministratori, che avranno accesso a strumenti per la gestione e creazione degli eventi. Attraverso un'interfaccia, gli amministratori potranno aggiungere nuovi eventi specificando il luogo, la tipologia, e i settori disponibili. Inoltre, sarà possibile definire il numero di posti per ciascun settore e impostare i prezzi relativi.

Modello di processo

Il modello di sviluppo scelto è quello di sviluppo incrementale. La funzionalità del sistema viene prodotta in piccoli incrementi, al fine di ottenere test e feedback più frequenti.



Seguiremo il modello a cascata con le sue varie fasi, comprese quelle di verifica e validazione tra ogni fase.

Organizzazione del progetto

L'organizzazione del team di sviluppo presenta le caratteristiche di un team agile di piccole dimensioni (3 persone). Sarà centrato sulla comunicazione tramite brevi canali e incontri in presenza, in cui i membri del team lavorano nella stessa stanza oppure in remoto, aggiornandosi e discutendo dei vari problemi relativi alla stesura e implementazione del progetto. All'interno del team non è presente una gerarchia di nessun tipo, ogni membro partecipa attivamente allo sviluppo del software e del progetto, assegnando a ciascuno varie attività. I 3 membri del team si tengono in costante aggiornamento per analizzare quotidianamente l'andamento del progetto.

Standard, linee guida, procedure

- Standard: i nomi delle classi sono Pascal Case (NomeClasseProva), i nomi dei metodi sono Camel Case (nomeMetodoProva), i nomi dei package sono Snake Case (nome_package_prova), i nomi delle variabili sono Camel Case (nomeVariabileProva).
- Linee guida: le priorità dei requisiti seguiranno le regole del MoSCoW (Must, Should, Could, Won't Have).
- Procedure: all'interno del team verranno individuati dei requisiti del progetto, realizzati i diagrammi UML, implementato il codice con successivo testing del sistema e verifica della qualità.

Attività di gestione

Utilizziamo GitHub per garantire la tracciabilità, il controllo sullo sviluppo del software e la gestione delle versioni del repository del progetto. Inoltre, disponiamo di un Trello, una lavagna digitale che consente a gruppi di persone di gestire visivamente qualsiasi tipo di progetto, flusso di lavoro o monitoraggio dei task. I vari task avranno date di scadenza che il team dovrà rispettare per ottenere aggiornamenti in tempo reale sull'avanzamento del progetto.

Rischi

L'unico rischio è quello di non consegnare il software in modo completo (con l'aggiunta di tutte le funzionalità secondarie) a causa di mancanza di tempo ed impegni al di fuori dell'ambito universitario da parte dei membri del team (rischio molto remoto).

Personale

Inizialmente tutti e 3 i membri del team si dedicheranno a tutte le fasi di sviluppo del progetto, per incrementare la coesione e comprensione del problema nel gruppo. Per la stesura del diagramma dei casi d'uso e il diagramma delle classi (e conseguente creazione del database) abbiamo deciso di dedicarci del tempo tutti contemporaneamente, al fine di avere un quadro chiaro del problema. Successivamente, a lavoro inoltrato, abbiamo stimato che potrebbero essere definiti dei ruoli più specifici per ogni attività/sezione riguardante l'implementazione del progetto.

Metodi e tecniche

Verranno definiti i seguenti UML, prima su un editor grafico in modo che possano essere modificati in attesa del modello finale che verrà definito tramite Papyrus:

- Use Case Diagram
- Class Diagram
- State Machine Diagram
- Sequence Diagram
- Communication Diagram/Timing Diagram
- Activity Diagram
- Component Diagram

Per quanto riguarda la fase di Verifica & Validazione, al fine che il sistema soddisfi i propri requisiti specifici (cioè che stiamo costruendo il sistema correttamente) e che il sistema risponda correttamente ai requisiti richiesti (cioè che stiamo costruendo il sistema giusto), verranno eseguiti test direttamente su Eclipse, eseguendo il codice dopo ogni fase e analizzando approfonditamente il corretto funzionamento.

Garanzia di qualità

La qualità del progetto sarà garantita da una completa documentazione della fase di testing, prendendo in considerazione 5 fattori di qualità chiave e usando i criteri di McCall:

- Correttezza: sarà garantita attraverso un approccio di sviluppo basato su continui test. Questo consentirà di avere un prodotto preciso e capace di soddisfare i vari obiettivi che sono stati imposti nel progetto.
- Affidabilità: il progetto sarà capace di rispondere nel tempo e a garantire lunghi periodi di utilizzo.
- Integrità: verrà garantito un elevato livello di protezione e integrità dei dati, al fine di avere una maggiore sicurezza e privacy.
- Usabilità: le interfacce utente saranno sviluppate pensando dapprima all'usabilità, attraverso un design minimal ed efficace, facilitando l'interazione da parte degli utenti, anche inesperti, che non conoscono appieno il funzionamento.
- Efficienza: l'obiettivo sarà ottimizzare al meglio le risorse disponibili per garantire un utilizzo e funzionamento fluido dell'applicazione, aumentando le performance.

Work Packages

Il carico di lavoro si dividerà in Documentazione, Codifica del Design, Codifica della parte logica e Codifica della parte di backend del DataBase.

Vengono utilizzati GitHub per la parte di codice, con assegnazione di vari issue in caso di problemi e Trello per l'assegnazione dei task ai vari membri del gruppo per l'organizzazione dell'intero progetto.

Risorse

Per la realizzazione del progetto saranno utilizzati i seguenti software e tools:

- Eclipse IDE: ambiente di sviluppo Java, per la stesura del codice
- Maven: strumento di *build automation* utilizzato prevalentemente nella gestione dei progetti Java
- Papyrus: tool utile per la creazione dei vari diagrammi UML richiesti in fase di progettazione del software
- SQLite: libreria software che permette di utilizzare database embedded, per gestire i dati e tabelle dell'applicazione

Per la stesura della documentazione e Project Plan è stato utilizzato Documenti Google.

Budget e programma

Non sono previsti costi di sviluppo, quindi non verrà definito nessun budget per il progetto. La consegna del Project Plan è prevista un mese prima dell'esame, ovvero il 17 dicembre, mentre il progetto può essere consegnato entro il 13 gennaio. Il tempo è di vitale importanza per la stesura del progetto complessivo: abbiamo stimato che ogni membro del team lavorerà all'incirca 50 ore ciascuno. Il tempo sarà suddiviso per la maggior parte nella progettazione e sviluppo del progetto (scrittura del codice e documentazione di esso), mentre il restante sarà utilizzato per studiare il problema in questione e per controllare/risolvere eventuali errori.

Cambiamenti

I cambiamenti che ogni membro del team apporta al progetto verranno gestiti discutendo e apportando piccole modifiche per ogni incremento iterativo. Per ogni cambiamento proposto da un membro del team e poi successivamente incorporato nel progetto, ne verrà tenuta traccia in questa apposita sezione.

Consegna

In data 18/11/2024 verrà consegnato il pdf del Project Plan, attraverso la creazione di una issue in GitHub. Invece il progetto completo verrà consegnato entro la deadline imposta del 13/01/2025 al Prof. Gargantini e alla Prof.ssa Bonfanti, attraverso la condivisione del repository su GitHub.