1. **SOFTWARE LIFE CYCLE**

Si è scelto di procedere seguendo uno dei metodi Agile: Extreme Programming. Questo promuove lo sviluppo iterativo ed incrementale organizzato in brevi cicli di sviluppo, detti interazioni.

Lo sviluppo è accompagnato dalla stesura di un piano di lavoro che viene continuamente aggiornato a intervalli brevi e regolari. In fase di pianificazione il team seleziona tutte le user story che possono essere completate nell’arco di tempo prestabilito. Infatti, così come per altri metodi Agile, lo sviluppo ha una durata breve che va dalle due alle quattro settimane.

Anche dal punto di vista dell’implementazione, Extreme Programming promuove la scrittura di soluzioni semplici che possono essere adattate e migliorate in un secondo momento attraverso refactoring o scrittura di componenti aggiuntive. Nell’Extreme Programming lo sviluppo prevede che tutti lavorino su tutto alternandosi durante alcune fasi, favorendo così il rendimento e mantenendo alto il livello di concentrazione.

Lo sviluppo tramite questo metodo prevede che ogni versione del codice rilasciata sia funzionante, non per questo viene definito come Test Driven Developement. Il sistema è verificato sia a livello di sistema, attraverso test basati su casi d’uso, sia a livello del singolo metodo. I test vengono ripetuti dopo ogni singola modifica per verificarne la stabilità.

La manutenzione, oltre a prevedere l’aggiunta di nuove componenti, prevede l’uso del refactoring e quindi la ristrutturazione del codice eliminando le parti superflue, semplificandone così la struttura.

---magari aggiungere grafico UML (MDA)

? 4- Configuration Management (Capitolo 4 libro)

-Deve usare un sistema per il configuration managment (gitHub) con i relativi comandi

-Deve dimostrare di aver usato issues, branches, pull requests, code rewies

1. **PEOPLE MANAGEMENT AND TEAM ORGANIZATION**

Il team è composto da 3 persone: Alexander Rubino (MAT. 1064467), Ibrahima Sarr (MAT. 1046446) e Beniamino Infante (MAT.1063452).

La struttura del team è semplice: tutti e tre i membri sono al centro del progetto e la coordinazione avviene mediante diretta supervisione.

Il lavoro viene suddiviso in maniera proporzionale, assegnando uno o due diagrammi UML per ogni singolo membro. Per la stesura del codice invece si è scelto di lavorare in coppia, alternandoci di volta in volta per favorire il rendimento e quindi il risultato finale.

La comunicazione è alla base del progetto, infatti sono fissate riunioni periodiche per riepilogare il lavoro svolto, discutere dei problemi incontrati e fissare nuovi obiettivi da raggiungere nel breve periodo.

1. **SOFTWARE QUALITY**

Il sistema deve essere conforme allo standard IEEE 9126.

Questo prevede i seguenti attributi:

-Funzionalità: il sistema deve essere sicuro e deve contenere le funzioni base descritte nei requisiti non funzionali.

-Affidabilità: il sistema deve avere una buona tolleranza agli errori e quindi non causare problemi gravi all’utente finale.

-Usabilità: il sistema deve essere facilmente utilizzabile dall’utente finale.

-Efficienza: il sistema non deve usare risorse eccessive. Deve essere ottimizzato per l’uso immediato.

-Manutenzione: il sistema deve essere regolarmente aggiornato risolvendo failure o bug incontrati e aggiungendo nuove funzioni richieste dall’utente.

Portabilità: il sistema deve poter girare su diversi OS.

1. **REQUIREMENTS ENGINIEERING**

-Deve contenere la specifica dei requisiti (esempio IEEE 830)

Requisiti Informali:

Si realizzi un software che consenta la gestione delle prenotazioni per le partite di calcetto e l’iscrizione a tornei di calcetto. Il sistema deve consentire agli utenti, per prenotare una singola partita, di scegliere la località e quindi i campi disponibili a seconda delle proprie richieste: tipologia della partita, il numero di partecipanti, l’età media dei partecipanti. Inoltre, si deve tener conto che spesso la disponibilità dei campi sarà limitata dalle condizioni meteorologiche, dalla disponibilità della struttura stessa e disponibilità del personale (addetti alla manutenzione del campo, custode del campo, arbitro).

In caso di mancata disponibilità l’utente potrà selezionare una delle località/strutture che sono rimaste disponibili.

Si distinguono due categorie di utenti: l’utente che si registra come singolo giocatore e l’utente che si registra come squadra.

Ciascun utente deve fornire dati utili come: nome, cognome, indirizzo di posta elettronica, numero telefonico e ruolo preferito (portiere, difensore, centrocampista e attaccante) per registrarsi al sistema. Deve anche indicare in quale località preferisce giocare e la tipologia di sport (calcio a 5, calcio a 8, calcio a 11).

Nel caso di utente squadra dovranno essere forniti i dati citati in precedenza insieme al nome della squadra e dell’allenatore ed una lista con nome, cognome e ruolo di ogni giocatore che compone la rosa. Ogni squadra prima di potersi iscrivere ad un torneo deve essere registrata nel sistema.

Entrambe le categorie di utenti, nel caso dell’utente squadra a registrarsi sarà il rappresentante, possono registrarsi ad una singola partita scegliendo la data e il luogo dove voler giocare. Inoltre, possono cancellare la prenotazione alla singola partita entro il limite previsto dal sistema altrimenti scatterà la sanzione.

Per l’iscrizione ai tornei, il sistema distingue tre differenti tipologie:

-Torneo di calcio a 5

-Torneo di calcio a 8

-Torneo di calcio a 11

Per ciascuno di questi è necessario che le squadre vengano formate 5 giorni prima poiché è necessario che venga raggiunto il numero minimo di squadre partecipanti per l’organizzazione del torneo. Nel caso in cui la squadra si dovesse ritirare dopo la scadenza dei tempi di iscrizione all’evento, verrà contattato tramite mail e dovrà pagare ugualmente la quota di partecipazione. Se questa azione si ripeterà due volte consecutivamente scatterà il ban dal sistema.

Alla ricezione di una prenotazione per una partita di calcetto il sistema invierà una mail all’utente con tutte le info di riepilogo sull’evento. Solo dopo la data di scadenza della prenotazione all’evento, il sistema invierà le informazioni relative ai giocatori che comporranno le due squadre. Nel caso dei tornei una volta raggiunto il numero di iscrizioni minimo il sistema fornirà il tabellone completo ed il calendario con date e squadre del torneo e ogni rappresentante di squadra verrà informato via e-mail. Dopo ogni partita verrà inviata una nuova mail con il tabellone e le date aggiornate.

Al termine della prenotazione e/o al termine dell’evento ciascun utente potrà fornire una recensione sul servizio offerto dal sistema, sui servizi legati al campetto (qualità manto erboso, arbitri, prezzo, spogliatoi).

Il sistema deve contenere:

* Una pagina di prenotazione dove si inseriscono i dati personali contenente anche le regole di utilizzo.
* Dovrà supportare il lavoro di amministrazione ed organizzazione delle strutture, dei partecipanti.
* Dovrà gestire l’elenco degli utenti con tutte le informazioni che possono risultare utili.
* Il sistema dovrà gestire gli eventi e il tabellone dei tornei
* Inizialmente viene fornito all’utente uno strumento di ricerca di eventi

Obiettivi e punti di vista:

Priorità dei requisiti

Tabella MoSCoW

1. **MODELING**

Inserire e descrivere i vari diagrammi UML

Parte di Uml

Architettura ->Diagramma delle componenti

1. **SOFTWARE ARCHITECTURE**

-Deve contenere la descrizione dell’architettura con almeno un paio di architectural views (per differenti punti di vista)

-Dovrebbe avere almeno una vista con connettori e componenti con la descrizione dello stile architetturale

Architettura Kruchten 4+1 view model?

1. **SOFTWARE DESIGN**

**Singleton pattern**

-Deve contenere una descrizione del design (mediante i diagrammi UmL)

-Potrebbe Calcolo Complessità?

-Dovrebbe contenere misurazione del codice, soprattutto se oo (con qualche metrica che abbiamo visto) Alcuni tools: STAN, JDepend, struture101 (funzionano solo con eclipse)

-Deve applicare un paio di design pattern visti a lezione della prof. Scandurra

1. **SOFTWARE TESTING**

-Può avere un documento di plan per l’attività di test

-Deve contenere dei casi di test di unità implementati con la loro descrizione nel documento

-Dovrebbe avere qualche misura di copertura per i casi di test

1. **SOFTWARE MAINTENANCE**

-Dovrebbe documentare alcune attività di refactoring che sono state fatte