

Piano di sviluppo del software (SDP)

Scopo del documento

Costruzione di un sistema domotico per gestire in maniera coordinata dispositivi (sensori e attuatori) in ambienti domestici, consentendo:

- Gestione di stanze e dispositivi
- Impostazione manuale di parametri ambientali
- Creazione e attivazione di scenari personalizzati/predefiniti
- Comunicazione event-driven tra componenti

Metodologia di sviluppo

Unified Process (UP) con 2 iterazioni totali:

Iterazione	Fasi UP coperte	Obiettivo
1	Inception + Elaboration + Construction (parziale)	Architettura stabile, casi d'uso core implementati
2	Construction (completamento) + Transition	Funzionalità complete, rilascio finale

Considerazioni:

- il progetto avrà solo 2 iterazioni. Quindi nella prima Iterazione verranno svolte Inception, Elaboration e parte della Construction. Nella seconda iterazione verranno verrà terminata la Construction e svolta la Transition

Documenti (deliverable) principali

- Documento di visione
- Documento dei requisiti

- Documento dei casi d'uso
- Documento di Architettura
- Codice sorgente
- Schema DB

Casi d'uso in Scope

UC	Nome
UC1	Gestione dispositivi (CRUD)
UC2	Impostazione parametri manuali
UC3	Creazione scenari personalizzati
UC4	Attivazione scenario
UC5	Gestione Stanze (CRUD)
UC6	Comunicazione e rilevazione guasti

Processo tecnico e strumenti

- **Linguaggi/Framework:** Java, JUnit
- **Gestione Versioni:** Git / GitHub
- **IDE:** VS Code, Eclipse.
- **Database:** MySQL
- **UI:** Java Swing
- **Build:** Maven

Pianificazione delle fasi e delle Iterazioni

Il primo step dello sviluppo UP di Smart DAB è stato quello di definire in modo chiaro e preciso le iterazioni, i tempi di sviluppo e soprattutto gli obiettivi da raggiungere.

Lo sviluppo del progetto si divide sostanzialmente in due iterazioni:

I Iterazione:

Definizione degli obiettivi del progetto, creazione dei documenti di progetto (documento di visione, dei requisiti, dei casi d'uso e i diagrammi UML associati) e sviluppo del primo sistema stabile per effettuare test di unità. Inoltre nel corso del progetto sono state effettuate diverse peer review per la discussione del codice scritto, al fine di progettare ogni componente in linea con tutte le idee e i flussi di sviluppo.

Timebox : ~1 mese (17 Dicembre - 25 Gennaio)

Milestones :

- Aggiunta prima versione dei documenti di progetto
- Sviluppo della prima versione dei casi d'uso UC1, UC2, UC3, UC4, UC5
- Implementazione della parte sulla comunicazione del caso d'uso UC6
- Aggiunti test di unità sui casi d'uso sviluppati

II Iterazione:

Aggiornamento dei documenti a seguito di change-requests, nate dallo sviluppo e dall'obiettivo di raggiungere tutti gli obiettivi entro i tempi fissati. Sono stati inoltre aggiunti test di unità e di integrazione per rifinire le funzionalità del sistema e per la preparazione alla presentazione del progetto.

Timebox : ~15 giorni (26 Gennaio - 12 Febbraio)

Milestones :

- Rifiniti casi d'uso della prima iterazione tramite test di unità e di integrazione
- Implementazione della rilevazione dei malfunzionamenti nel caso d'uso UC6
- Aggiornati documenti post-sviluppo
- Aggiunta presentazione del progetto

Iteration Assessment

Durante la prima iterazione, l'obiettivo principale è stato la definizione dei requisiti e la progettazione dell'architettura di sistema. E' stata completata la creazione dei documenti:

- Analisi dei requisiti: documento di visione, documento dei requisiti, documento dei casi d'uso e glossario
- Modellazione: diagramma di dominio, diagramma dei casi d'uso
- Progettazione: diagramma delle classi e diagrammi di sequenza

Oltre ai documenti, l'iterazione ha prodotto un prototipo eseguibile del sistema. E' stato completato lo sviluppo dei casi d'uso definiti nella fase di pianificazione, la milestone è stata raggiunta.

La seconda iterazione è stata focalizzata sulla revisione di quanto sviluppato. Le attività principali sono state:

- revisione dei documenti: aggiornamento della documentazione a seguito di change requests specifiche in modo da garantire allineamento tra requisiti e implementazione.
- test: esecuzione di test sul codice per identificare bug
- refactoring: modifica e ottimizzazione di alcune funzionalità esistenti sulla base dei risultati dei test

Anche per questa seconda iterazione la milestone prefissata è stata raggiunta. Il superamento dei test conferma la stabilità del sistema.