



SWE16

GITKRAFFEN

Verbale Esterno

Gruppo GitKraffen – Progetto IronWorks
gitKraffen.swe16@gmail.com

Versione	Unica
Redazione	Elia Rebellato
Verifica	Daniel Rossi
Approvazione	Mattia Poloni
Uso	Interno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo GitKraffen

Descrizione

Questo documento riassume il colloquio del gruppo GitKraffen col Prof. Cardin tenuto il giorno 22 Maggio 2018.

1 Informazioni Generali

1.1 Informazioni Incontro

- **Luogo:** Padova, Aula 2BC30 Torre Archimede;
- **Data:** 22 Maggio 2018;
- **Ora:** 18:00 - 18:20;
- **Partecipanti del gruppo:** Mihai Eni, Federica Ramina, Elia Rebellato, Daniel Rossi, Mattia Poloni.
- **Partecipanti esterni:** Prof. Riccardo Cardin.

1.2 Argomenti trattati

L'incontro si è tenuto per discutere delle segnalazioni fatte dal Prof. Riccardo Cardin in seguito la Revisione di Progettazione sul documento Analisi dei Requisiti v2.0.0 ed inoltre per risolvere dei dubbi riguardo il diagramma delle classi.

2 Domande e risposte

2.1 Domande e risposte sull'Analisi dei Requisiti :

- **Come mai nel caso d'uso UC0, ci è stato segnalato che continua ad essere troppo legato all'implementazione?**

R: E' stato segnalato in quanto la visualizzazione della pagina iniziale non è una funzionalità del sistema. Ciò che dovete individuare è cosa il vostro prodotto offrirà agli utenti, ovvero corrispondono alle funzionalità che avete descritto nei sottocasi, ma non può avere una funzionalità di visualizzazione della pagina iniziale;

- **Per quanto riguarda i casi d'uso 2.1 e 2.2, è stato segnalato che probabilmente sono in una relazione di gerarchia con UC2. Questo in quanto è possibile effettuare una delle due funzioni, dunque rappresenta come una scelta dell'utente da dove caricare il file?**

R: Esattamente. Inoltre definire il caso 2.1 con "Apertura file" è scorretto, in quanto non rappresenta una funzionalità. Più corretto sarebbe il definirlo come "Caricamento del file";

- **Come mai ci è stato segnalato che UC3 non può contenere sia l'inserimento che la modifica del nome di un progetto?**

R: La creazione di un progetto e la modifica di un progetto sono casi d'uso separati, perchè agiscono su pre-condizioni differenti. Il caso d'uso modifica nome progetto lo eliminerei, in quanto poter modificare più volte il nome del progetto non è molto appropriato.

- **Per quanto riguarda l'UC4 dobbiamo solo raggruppare le funzionalità?**

R: Sì, in quanto attualmente risulta troppo caotico. Quando delle funzionalità possono essere raggruppate conviene farlo per una maggior chiarezza, ad esempio si può mettere:

1. gestione attore;
2. gestione relazione;

3. gestione entity.

- **Come mai sono da rivedere i titoli dei casi d'uso UC4.6.2 e UC4.6.2.3?**

R: In quanto da una visualizzazione ci si aspetta dei sottocasi che indicano meglio cosa si visualizza, non dei sottocasi che trattano di modifiche.

2.2 Domande e risposte sul diagramma delle classi:

- **Per quanto riguarda il pattern MVC con backbone dobbiamo astrare per le classi come se fossimo un UML per java o c++?**

R: il fatto che non ci sia scritto class da nessuna parte non vuol che non esista un concetto di contenitore, cioè dovete vedere un tipo come un contenitore con le funzione e dei dati che operano su queste funzioni;

- **Quello che ci confonde è che non possiamo fare una corrispondenza 1 a 1, quindi come si può fare?**

R: non si astrae, come capita spesso anche in java, l'UML non fa il binding su nessun linguaggio, in quanto non esiste una regola 1 a 1 ma dovete deciderle voi. Quindi un volta che avete definito delle regole per passare da UML a codice. Il modo in qui state facendo il diagramma per adesso è corretto;

- **Nel diagramma teniamo la struttura con observer oppure si lavora con events come dice backbone?**

R: lavorate pure come dice backbone quindi implementate events. Inoltre events probabilmente non è altro che un observer che viene implementato dalla libreria, quindi va benissimo seguire quello che dice la documentazione di backbone.