



Verbali Esterni

Gruppo SWEet BIT – Progetto SWEDesigner

Informazioni sul documento

| | |
|----------------------|---|
| Versione | 1.3.0 |
| Redazione | Santimaria Davide Massignan Fabio |
| Verifica | Massignan Fabio Bodian Malick |
| Approvazione | Pilò Salvatore |
| Uso | Esterno |
| Distribuzione | Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo SWEet BIT Zucchetti S.p.A. |

Descrizione

Questo documento traccia i verbali di tutte le riunioni esterne, con il *Committente_G* ed il *Proponente_G*, del gruppo SWEet BIT.

Versioni del documento

| Versione | Data | Persone coinvolte | Descrizione |
|----------|------------|----------------------|--|
| 1.3.0 | 2017/03/22 | Pilò Salvatore | Approvazione Documento |
| 1.2.0 | 2017/03/19 | Bodian Malick | Verifica Documento |
| 1.1.1 | 2017/03/16 | Massignan Fabio | Stesura Riunione Esterna 2017/03/15 |
| 1.0.0 | 2017/03/16 | Santimaria Davide | Creazione struttura documento |

Indice

| | |
|---|----------|
| 1 Riunione 2 | 3 |
| 1.1 Informazioni sulla riunione | 3 |
| 1.2 Domande e Risposte | 3 |

1 Riunione 2

1.1 Informazioni sulla riunione

- **Data:** 15/03/2017;
- **Luogo:** Zucchetti - sede di Padova , via Giovanni Cittadella 7;
- **Ora:** 11:30;
- **Durata:** 20 min;
- **Argomento:** Consulenza sui *diagrammi_G*;
- **Partecipanti Interni:** Santimaria Davide - Massignan Fabio - Salmistraro Gianmarco - Bodian Malick - Pilò Salvatore - Bertolin Sebastiano;
- **Partecipanti Esterni:** Piccoli Gregorio.

1.2 Domande e Risposte

- Per la gestione tra i *diagrammi_G* delle *classi_G* e dei *metodi_G*, abbiamo pensato di mettere a disposizione dell'*utente_G* una finestra principalmente divisa in 3 parti. La prima parte comprende gli strumenti necessari a disegnare i vari *diagrammi_G*, la seconda finestra è quella principale e consiste nello schema delle *classi_G* e, in caso di doppio click sul *metodo_G* di una *classe_G*, si aprirà un *diagramma delle attività_G*, col quale l'*utente_G* potrà creare e definire il corpo del *metodo_G*. Nella terza finestra, l'*utente_G* ha a disposizione un *diagramma delle attività_G* in sola lettura, sul quale è possibile visualizzare il flusso dell'intero progetto. Come le sembra la nostra soluzione?

Questa disposizione è apprezzata, l'importante è non saltare il *diagramma_G* delle *classi_G* e trattare il *diagramma delle attività_G* in modo da poter generare il *codice_G*.

- Data la scelta del *dominio_G* dei giochi da tavolo, possiamo aggiungere dei *template_G* di *classi_G* e funzioni legate al *dominio_G*?

L'*UML_G* non ha un grande supporto dei *template_G*, solitamente si usano le collaborazioni, che son un ente che descrivono un insieme statico di relazioni tra istanze, e i ruoli che queste istanze svolgono in queste relazioni. Questa soluzione è abbastanza scomoda; suggerisco di approfondire i *pattern_G* che utilizza l'*UML_G* per capire da sé che inizialmente l'*UML_G* è nato per disegnare *classi_G* e quindi con l'uso dei *pattern_G* c'è contrasto. Ci son due rappresentazioni di *pattern_G* quella della sua definizione e quella del suo uso; implementare la possibilità all'*utente_G* di crearsi i propri *template_G*. Quando l'*utente_G* creerà il suo *template_G*, le *classi_G* che

ne fanno parte dovranno avere un'etichetta che ne definisce il ruolo e la funzionalità del cambio del nome; altrimenti se in un *diagramma_G* si richiama più volte lo stesso *template_G* ciò darà luogo ad errori visto che il nome è un attributo univoco. Un esempio può essere un videogioco di calcio, dove per rappresentare la testa dei giocatori si usa un *pattern_G*, che a sua volta può essere un contenitore di altri *pattern_G* come occhi, e naso. Già questo *pattern_G* si può utilizzare 22 volte per il numero di giocatori in campo ed ognuno di essi avrà delle caratteristiche differenti, ma seguono tutti lo stesso modello.

- **Il passaggio tra *codice_G* e *Diagramma_G* è un requisito richiesto?**

No, questa caratteristica non è necessaria. Si potrebbe valutare eventualmente la sincronizzazione, o meglio dare la possibilità a due persone di lavorare allo stesso *diagramma_G*, ma anche questa è una caratteristica che può essere inserita tra i requisiti opzionali. Ci si potrebbe orientare su un disegnatore collaborativo, che supporti *JSON_G* e librerie come *diff-match-Patch_G*, così da permettere di far lavorare in maniera collaborativa più utenti, grazie allo scambio di *file_G* in formato *JSON_G*; resta comunque una caratteristica che diventa un progetto all'interno del progetto attuale e quindi, come già detto, è un requisito opzionale.

- **È obbligatorio fornire l'auto compilazione del *codice_G*?**

Se si intende che il *codice_G* generato debba compilare questo è banalmente ovvio. La compilazione automatica sarebbe gradita.

- **Dobbiamo aggiungere la presenza di un *utente_G* amministratore?**

Credo che la mole di lavoro sia sufficiente e che quindi aggiungere nuove funzionalità al momento si possa escludere, basta concentrarsi sui requisiti già definiti. Avete accennato alla possibilità di aggiungere dei gruppi di lavoro, ma preferisco che vi concentrate sull'inserimento dei *template_G* anziché di questa funzionalità.