



# Verbali Esterni

*Gruppo SWEet BIT – Progetto SWEDesigner*

## Informazioni sul documento

<b>Versione</b>	1.3.0
<b>Redazione</b>	Santimaria Davide Massignan Fabio
<b>Verifica</b>	Massignan Fabio Bodian Malick
<b>Approvazione</b>	Pilò Salvatore
<b>Uso</b>	Esterno
<b>Distribuzione</b>	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<sub>G</sub>* ed il *Proponente<sub>G</sub>*, del gruppo SWEet BIT.

## Versioni del documento

Versione	Data	Persone coinvolte	Descrizione
1.3.0	2017/03/22	Pilò Salvatore	Approvazione Documento
1.1.0	2017/02/27	Massignan Fabio	Verifica Documento
1.0.1	2017/02/24	Santimaria Davide	Stesura Riunione Esterna 2017/02/23
1.0.0	2017/02/23	Santimaria Davide	Creazione struttura documento

## Indice

<b>1 Riunione 1</b>	<b>3</b>
1.1 Informazioni sulla riunione . . . . .	3
1.2 Domande e Risposte . . . . .	3

## 1 Riunione 1

### 1.1 Informazioni sulla riunione

- **Data:** 23/02/2017;
- **Luogo:** Zuccheti - sede di Padova , via Giovanni Cittadella 7;
- **Ora:** 16:30;
- **Durata:** 90 min;
- **Argomento:** Chiarimenti sui requisiti del *capitolato<sub>G</sub>*;
- **Partecipanti Interni:** Santimaria Davide - Massignan Fabio - Salmistraro Gianmarco - Bodian Malick - Pilò Salvatore - Bertolin Sebastiano;
- **Partecipanti Esterni:** Piccoli Gregorio.

### 1.2 Domande e Risposte

- **Come dev'essere la qualità del *codice<sub>G</sub>* generato dai *diagrammi<sub>G</sub> UML<sub>G</sub>*?**  
Per rispondere al meglio alla domanda, ci si vuole dapprima focalizzare su un altro punto, ovvero il perché sia stato suggerito il tema dei giochi da tavolo in questo *capitolato<sub>G</sub>*.

Ebbene nel *dominio<sub>G</sub>* dei giochi da tavolo, la ripetitività è molto elevata; basti pensare al gioco della Dama o degli Scacchi, entrambi condividono la stessa scacchiera di gioco, anche il Monopoli ha una scacchiera, che si diversifica per colori e tipi di caselle, ma è pur sempre una scacchiera. Focalizzarsi su un specifico ambito, in questo caso i giochi da tavolo, aiuta a migliorare la qualità del *codice<sub>G</sub>* (vista anche la possibilità del suo riutilizzo), in quanto se si decidesse di rappresentare ogni contesto, risulterebbe difficoltoso generare del *codice<sub>G</sub>* che si adatti al meglio in ogni situazione.

Sfruttare il fattore del riutilizzo sicuramente aiuta, ma la scelta che incide è il modo in cui si sceglie di disporre i *diagrammi<sub>G</sub>*, ovvero se visualizzarli per *layer<sub>G</sub>*, suddividendoli per categorie ad esempio, oppure disporre il tutto su un unico *layer<sub>G</sub>* ed utilizzare un sistema di evidenziazione, *zoom<sub>G</sub>* o altre tecniche. Tale scelta è importante perchè riflette quanto più complesso e specifico potrà essere il progetto che l'*utente<sub>G</sub>* vorrà realizzare, e da questo ne deriva anche la qualità del *codice<sub>G</sub>*. La vera difficoltà sta nel generare *codice<sub>G</sub>* dai *diagrammi<sub>G</sub>* che rappresenteranno i *metodi<sub>G</sub>*, per essi si suggerisce un approccio utilizzando il *diagramma delle attività<sub>G</sub>*.

- **Nel nostro  $designer_G$  dobbiamo includere dei  $template_G$ ?**

Si, a patto che venga scelto un  $dominio_G$  su cui il  $designer_G$  si basi.

Mettere a disposizione ad esempio una scacchiera 8x8, evitando che l' $utente_G$  finale debba crearsela da zero. Ovviamente i giochi da tavolo non son l'unico  $dominio_G$  su cui ci si può basare, la scelta potrebbe ricadere su altri settori, ma tale scelta deve esser fatta per poter avere dei  $diagrammi_G$  e quindi del  $codice_G$ .

Il  $dominio_G$  fin ora citato è consigliato perchè offre molti approfondimenti sull'adattamento in  $diagrammi_G$  ed inoltre garantirà una fase di testing più piacevole.

- **Se viene modificato il  $codice_G$  generato, il  $diagramma_G$  deve aggiornarsi anch'esso?**

È un aspetto sicuramente interessante se si riuscisse ad implementare. Solitamente dopo la creazione dei  $diagrammi_G$  ed il rispettivo  $codice_G$ , le successive modifiche vengono apportate solo al  $codice_G$ ; in quanto il  $diagramma_G$  ha lo scopo principale di fungere da linea guida, rappresentando solo le  $classi_G$  principali, tralasciando nella visualizzazione quelle di supporto. Rappresentare tutte le  $classi_G$  che costituiscono il progetto, potrebbe ridurre la leggibilità del  $diagramma_G$ ; una soluzione potrebbe essere quella di celare o inserire in un  $layer_G$  diverso le  $classi_G$  di supporto.

Visualizzare solo lo scheletro della  $classe_G$  senza l'implementazione dei  $metodi_G$  può esser una soluzione alla domanda posta.

- **L' $Applicazione_G$  deve essere solo  $desktop_G$  o deve essere anche una  $Web-App_G$ ?**

È preferita la  $Web-App_G$ , ma la scelta non è vincolante. Lo scopo principale è entrare nell'ottica di fare un progetto usando molto i  $diagrammi_G$ .

- **Il  $codice_G$  prodotto deve esser in formato  $Java_G$  o  $JavaScript_G$ ?**

Nella fase di disegno dei  $diagrammi_G$ , con particolare riferimento a quelli dei  $metodi_G$ , si deve procedere in modo astratto, ovvero tracciando solo l'algoritmo necessario al  $metodo_G$  in esame. Procedendo con quest'ottica si può generare  $codice_G$  in entrambi i linguaggi. Proseguire in quest'ottica di pensare per specifiche risulta interessante ed una buona sfida, ma se si decidesse di pensare per programma, il consiglio è quello di procedere inizialmente con  $Java_G$ , che è un linguaggio più controllato e verificato.

- **L' $Applicazione_G$  dovrà esser disponibile su di un  $Server_G$  oppure è sufficiente in locale?**

Non è necessario l'acquisto di uno spazio su cui ospitare l' $Applicazione_G$ , eventualmente si possono utilizzare dei  $Server_G$  gratuiti a tempo limitato come ad esempio

Amazon, Heroku.

- **Considerando che il numero di gruppi che aderiscono a tale  $capitolato_G$  è aumentato, alcuni dei requisiti opzionali son diventati obbligatori?**

I requisiti opzionali rimangono invariati, un aspetto su cui focalizzarsi è lo studio dell' $UML_G$ . Il progetto consiste nel creare un disegnatore che abbia il  $diagramma_G$  della  $classe_G$  con i rispettivi  $diagrammi_G$  dei  $metodi_G$ ; la soluzione di come rappresentare il collegamento tra questi  $diagrammi_G$ , è sicuramente il punto su cui ci si deve concentrare, proponendo anche modelli ibridi dei  $diagrammi_G$ .

- **Ci devono essere dei controlli nella costruzione del  $diagramma_G$ ? E come devo restituire il  $codice_G$  generato?**

Ciò che viene restituito può essere un pacchetto contenente i  $file_G$  oppure una visualizzazione testuale del  $codice_G$ ; l'importante è che venga visualizzato il risultato che l' $utente_G$  si aspetta. Per quanto riguarda i controlli, sono un lavoro di cortesia, evitare che l' $utente_G$  durante la fase di "disegno" faccia degli errori come ad esempio dimenticarsi il nome della  $classe_G$  è sicuramente apprezzato. Ovviamente se si desse la possibilità di aggiungere nel  $diagramma_G$  un rettangolo dove si possa aggiungere del  $codice_G$  manualmente, in quella situazione risulterà impegnativo gestire automaticamente l'errore.