



# Verbale 12/03/2018

*WarMachine – Progetto IronWorks*

warmachine.swe@gmail.com

## Informazioni sul documento:

<b>Versione</b>	-
<b>Data di creazione</b>	12/03/2018
<b>Redazione</b>	Cisternino Nicola
<b>Verifica</b>	Bernucci Riccardo
<b>Approvazione</b>	Coletti Andrea
<b>Uso</b>	Esterno
<b>Distribuzione</b>	<i>Prof. Vardanega Tullio, Prof. Cardin Riccardo, Zucchetti s.p.a, gruppo WarMachine</i>



## 1 Informazioni Generali

- **Luogo incontro:** Padova (PD), Via Giovanni Cittadella 7, Sede Zucchetti;
- **Data Incontro:** 12/03/2018;
- **Ora inizio incontro:** 14:30;
- **Ora fine incontro:** 16:10;
- **Durata incontro:** 1 ora e 40 minuti;
- **Oggetto<sub>G</sub>:** Individuazione di un canale di comunicazione tra il Team<sub>G</sub> e l'Azienda e chiarimenti sul capitolato<sub>G</sub>;
- **Segretario:** Cisternino Nicola;
- **Partecipanti:** Bragnolo Leonardo, Cisternino Nicola.

## 2 Riassunto dell'incontro

Conclusioni preliminari primo incontro:

- **VE\_12032018\_1 - Canale di comunicazione con committente<sub>G</sub>.** Tramite email oppure richiesta incontro diretto fino a consegna del primo prototipo. Non concesso un canale di comunicazione come Slack<sub>G</sub>, Telegram o altri canali poco formali;
- **VE\_12032018\_2 - Punto 8 requisiti obbligatori.** Le entità consistenti si categorizzano in due tipi: quelle che possono essere supportate da un file Json<sub>G</sub> o xml<sub>G</sub>; sostanzialmente un singleton, i parametri di configurazione, l'utente del sistema<sub>G</sub>, un oggetto. E poi ci sono altre entità che sono quelle degli archivi (l'archivio lo si vuole leggere, scrivere, interrogare). Necessità di aggiungere una Search all'archivio;
- **VE\_12032018\_3 - Le interfacce** si dividono in due categorie: le interfacce grafiche (che interagiscono con l'utente) e quelle tra web server, o rest, machine to machine;
- **VE\_12032018\_4 -** Nei canali tra verbi-nomi si può estendere il canale con controlli sulla veridicità dei collegamenti tra controller e boundary. Il problema principale è che i programmatori danno per scontato che ciò che arriva dal server sia stato validato dal client<sub>G</sub>; invece questo è il punto di massimo attacco dagli hacker. E' importante fare sia un **controllo server-side** che **client-side** per avere una maggior sicurezza;
- **VE\_12032018\_5 -** Questo del "**robustness diagram**" è un sistema di analisi che prosegue nel diagramma delle classi. Quindi si può aumentare. E' come se stesimo passando **da un'analisi a una progettazione**. Questo avviene attraverso l'**aggiunta di dettagli**. E' una proprietà importante, che si chiama singleness, cioè non ho un salto (un grosso difetto dell' Uml<sub>G</sub> è che dai diagrammi d'uso ai

diagrammi delle classi c'è un "salto" o "stacco"), quello che è interessante, quindi, è garantire una continuità tra i due tipi di diagrammi. Non è il caso dei robustness diagram. Con un aumento delle cose che vengono dette in questo diagramma possiamo guadagnare una certa **continuità** e passare dall'analisi alla progettazione e arrivare alla produzione del codice. Il Robustness diagram è inteso inizialmente come strumento di analisi: si chiede di ampliare, tramite estensioni, le solite opzioni di analisi in modo tale da colmare il "salto" tra casi d'uso<sub>G</sub> e diagramma delle classi, guadagnando, quindi, continuità tra questi;

- **VE\_12032018\_6** - Le "**Regole**" del punto 8 dei requisiti obbligatori vanno intese come metodi necessari **che limitano la modalità di creazione di boundary e controller**. Esempio: se ho un metodo che nel codice auto-generato ha bisogno di essere letto allora una delle "regole" mostrerà che ci sarà bisogno di un metodo read per leggere i dati;
- **VE\_12032018\_7** - **Librerie utili** per disegnare diagrammi sono: jointjs<sub>G</sub> e cytoscape<sub>G</sub> con ambiente di programmazione backbone<sub>G</sub>. Backbone, angular<sub>G</sub> e react<sub>G</sub>: ricercare materiale;
- **VE\_12032018\_8** - **Tecnologie web da utilizzare**: meglio utilizzare Html5<sub>G</sub> e Css3<sub>G</sub>;
- **VE\_12032018\_9** - **Codice di manutenzione delle tabelle**: attraverso il tracciato record e le chiavi primarie si dovrà trovare la procedura adatta per la manutenzione. **hibernate<sub>G</sub>** è utile per questo principio, sarà possibile sostituire il requisito obbligatorio con Hibernate;
- **VE\_12032018\_10** - **Scopo del progetto<sub>G</sub>**: dare la possibilità di visionare la parte centrale tra logica e interfacce, cioè la parte che permette di guardare come avviene il passaggio dei dati nella **logic view**. Fare tutti i check sia sul client che sul server (per autenticità del client) in modo tale da verificare l'integrità dei dati.
  1. Possibilità di estendere dividendo parte server da parte client;
  2. Possibilità di gestire la **veridicità dei dati** sul canale invece che sul server, sul server ci potrebbe essere la possibilità di utilizzare un archivio di supporto. Derivo le condizioni dal programma scritto in Javascript<sub>G</sub> sul client in modo da aumentare la sicurezza;
  3. Possibilità di utilizzare Hibernate su entità;
  4. Utilizzare un **controller per ogni funzione**. Esempio: siamo in un hotel, hanno prenotato una serie di camere. Un controller gestisce tutte le camere che hanno prenotato, e un altro gestisce una cella alla volta dell'array arrivato al primo controller;
- **VE\_12032018\_11** - Formalismo di **Warnier-Orr**: Possibilità di utilizzare Json al posto di Warnier-Orr, è un linguaggio utilizzato dall'utente stesso per specificare l'oggetto passato tra le verb-noun;



- **VE\_12032018\_12 - Regole di calcolo e manutenzione:** Si effettuano controlli anche sui dati inseriti su un'interfaccia:
  - Rejector: controlla e rifiuta;
  - Calculator: controlla ed esegue operazione di calcolo;
  - Projector: controlla e proietta i dati alla prossima destinazione.
- **VE\_12032018\_13 - Non interessa** la parte di logica, ma più che altro l' **interazione tra oggetti**.

### 3 Riepilogo

Codice ID	Decisione
VE_12032018_1	Canale di comunicazione con committente
VE_12032018_2	Chiarimenti sul punto 8 dei requisiti obbligatori
VE_12032018_3	Interfacce
VE_12032018_4	Controlli tra control e boundary
VE_12032018_5	Robustness diagram
VE_12032018_6	"Regole" del punto 8 dei requisiti obbligatori
VE_12032018_7	Librerie utili per disegnare diagrammi
VE_12032018_8	Tecnologie web da utilizzare
VE_12032018_9	Codice di manutenzione delle tabelle
VE_12032018_10	Scopo del progetto
VE_12032018_11	Chiarimenti sul formalismo Warnier-Orr
VE_12032018_12	Regole di calcolo e manutenzione
VE_12032018_13	Interazione tra oggetti

**Tabella 1:** Tracciamento decisioni 12-03-2018