SWEetBIT

SWEDesigner

editor di diagrammi UML con generazione di codice

# Studio di fattibilità

*Redattori:*
Salvatore Pilò

*Approvazione:*
NomeCognome
*Verifica:*
NomeCognome


Versione 1.0.0

27 febbraio 2017

# Indice

# Diario delle modifiche

| Versione | Data | Autore | Descrizione |
|---|---|---|---|
| 1.0.0 | 2017-02-27 | *Analista* Salvatore Pilò | Creazione scheletro del documento, stesura introduzione e analisi capitolato |

Tabella 1: Diario delle modifiche

# Introduzione

## 1.1 Scopo del documento

Lo scopo di questo documento è quello di descrivere le motivazioni dietro la scelta del capitolato C6, SWEDesigner, da parte del gruppo SWEtBIT.

## 1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazone di una Web App che fornisca all'utente un UML Designer con il quale riuscire a disegnare correttamente diagrammi delle classi e descrivere il comportamento dei metodi interni alle stsse attraverso l'utilizzo di -da decidere il tipo di schema-. La Web App permetterà all'utente di generare codice Java o Javascript dal diagramma disegnato ed eventualmente andare a ritoccarne il risultato al fine di ottenere un codice eseguibile, funzonante e funzionale.

## 1.3 Glossario

Con lo scopo di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v1.0.0*. In esso saranno definiti, in modo chiaro e conciso i termini che possono causare ambiguità o incomprensione del testo.

## 1.4 Riferimenti

### 1.4.1 Informativi

- **Capitolato d'appalto C1:** APIM: An API Market Platform
  `http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C1.pdf`
- **Capitolato d'appalto C2:** AtAVi: Accoglienza tramite Assistente Virtuale
  `http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C2p.pdf`
- **Capitolato d'appalto C3:** DeGeOP: A Designer and Geo-localizer Web App for Organizational Plants
  `http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C3p.pdf`
- **Capitolato d'appalto C4:** eBread: applicazione di lettura per dislessici
  `http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C4p.pdf`
- **Capitolato d'appalto C5:** Monolith: an interactive bubble provider
  `http://www.math.unipd.it/~tullio/IS-1/201/Progetto/C5p.pdf`
- **Capitolato d'appalto C6:** SWEDesigner: editor di diagrammi UML con generazione di codice
  `http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6p.pdf`

### 1.4.2 Normativi

- **Norme di progetto:** *Norme di progetto v1.0.0*

# Scelta del capitolato C6

## 2.1 Descrizione del capitolato

Il capitolato C6, proposto dall'azienda *Zucchetti*, propone lo sviluppo di una Web App costituita da un designer di **UML** che utilizzi sia gli schemi tipici del linguaggio, come ad esempio il diagramma delle classi, sia alcuni ibridi ideati appositamente per lo scopo. Dal diagramma UML prodotto sarà possibile generare automaticamente del codice *Java* e/o *Javascript* chee può e deve essere modificabil dall'utilizzatore. In particolare è rchiesta una certa coerenza fra il codice scritto e i diagrammi presenti all'interno del designer. Le richieste principali del capitolato sono le seguenti:

- La trasformazione degli **UML** in linguaggio *Java* e/o *Javascript*;
- L'utilizzo di strutture tipiche del linguaggio **UML**;
- L'utilizzo di **TOMCAT** o *Node.js* per quanto riguarda il lato server;
- Il corretto funzionamento del prodotto finale su browser supportanti *Html 5.0* e *CSS 3*;

## 2.2 Dominio applicativo

Il capitolato pone come obbiettivo quello di creare uo strumento che possa automatizzare, nei limtii del possibile, il processo di generazione di codice. Negli ultimi anni si sente sempre di più l'esigenza di sviluppare *software* in tempi esigui e spendendo meno risorse possibili nella mano d'opera. Oltre a tutto questo si sente la necessità di avere davanti del codice quanto più pulito possibile da errori umani, pertanto l'esigenza di un tool in grado di automatizzare questo processo macchinoso, rendendo meno influente l'azione umana (e relativi errori) sul prodotto finale.
Nella pratica un tale sistema sarebbe impossibile da realizzare per via della mole di varibili in gioco, pertanto si deve provare a ridimensionare il problema ponendolo all'internodi un dominio specifico. In questo caso il dominio indicato dal proponente è quello dei giochi da tavolo -inserire altri domini qualora volessimo- così da ridimensionare notevolmente il problema: si tratta di un dominio molto specifico in cui è più "semplice" riuscire a generare del codice adatto alla situazione molto più particolare. Ad esempio è noto a tutti che un gioco da tavolo mette sempre a disposizione una plancia di gioco, la quale, nonostante ne esistano varie versioni, ha sempre degli elementi fissi che possono essere utilizzati a nostro vantaggio.

## 2.3 Dominio tecnologico

Vista la natura di Web App del capitolato e sopratutto alla luce dei requisiti richiesti dal proponente si è reso necessario uno studio approfondito in diversi campi:

- **Server TOMCAT:** conoscenza delle strumentazioni offerte da questa particolare tecnologia Apache con conseguenti pro e contro del caso.
- **Node.js:** conoscenza di questa piattaforma: in particolare si rendono necessarie le conoscenze della sua offerta e della possibili applicazioni all'interno del progetto.
- **JVM:** conoscenze di base del funzionamento della macchina virtuale di Java.
- **Java/Javascript:** conoscenza abbastanza approfondita dei due linguagg necessaria per la generazione del codice automatico a partire dagli **UML**.

- **Diagrammi UML:** conoscenza dei principali schemi utilizzati all'interno dello standard **UML**.
- **Meteor:** conoscenza basilare della piattaorma per agevolare la scrittura del lato client della Web App.

## 2.4 Criticità potenziali e costi

Tutte le tecnologie richieste per la realizzazione del progetto sono gratuite quindi non è richiesto un'impegno monetario per utilizzarle, tuttavia essendo in gran parte nuove per i membri del gruppo l'acquisizione delle competenze necessarie richiederà un investimento non banale in termini di tempo.

In maniera più specifica le tecnologie che possono ssere fonti di forti criticità sono le seguenti:

- **Diagrammi UML:** nessun componente del gruppo ha mai avuto a che fare con la progettazione di diagrammi UML, salvo che durante i corsi didattici ancora in corso. Lo studio approfondito di tale strumenti è fondamentale per la realizzazione del progetto.
- **Java/Javascript:** il gruppo possiede una conoscenza piuttosto generale dei linguaggi in questione. Si rende quindi necessario un approfondimento di tali conoscenze.
- **Node.js/TOMCAT:** nessun componente del gruppo ha avuto a che fare con tali tecnologie per lo sviluppo del lato server, si rende pertanto necessaria una conoscenza generale per la scelta della tecnologia da adoperare da approfondire maggiormente in seguito.
- **Meteor:** nonostante i compoenti del gruppo abbiano una conoscenza piuttosto basilare e generica della piattaforma è necessario uno studio più approfondito della stessa.

## 2.5 Analisi del mercato e benefici

Attualment sul mercato non sono disponibili strumenti di questo genere di si offrono di generare del codice in maniera automatizzata. I pochi esempi che possiamo ritrovare prevedono un sistema poco funzionale di drag-and-drop che genera del codice non sempre ottimale. Oltre a questo si sente molto l'esigenza di un mabiente che possa diminuire drasticamente i tempi di sviluppo software all'interno di un'azienda permettendo quindi al progetto di rispondere ad una richiesta piuttosto importante all'interno del mercato.
Il rilascio su licenza MIT permetterà infine una potenziale rapida crescita del progetto grazie al possibile apporto della comunità.

## 2.6 Considerazioni e valutazioni finali

Conseguentemente alle considerazioni esposte nelle sezioni precedenti il gruppo ha definito un insieme di aspetti positivi e negativi del capitolato:

### 2.6.1 Aspetti positivi

- **Interesse:** i componenti del gruppo hanno manifestato un interesse elevato nei confronti del dominio applicativo e delle tecnologie necessarie allo sviluppo, soprattutto per via dell'enorme potenzialità creativa dello stesso;
- **Novità:** il prodotto rappresenta un'interessante novità per il mercato che ha stimolato particolarmente i componenti del gruppo;
- **Esperienza:** lo sviluppo del prodotto permetterà ai membri del gruppo di acquisire competenze utili nel proseguimento della carriera grazie a tecnologie come *Node.Js*.
- **Licenza:** il rilascio del prodotto con licenza MIT fornisce interessanti prospettive future di utilizzo e sviluppo;

### 2.6.2 Aspetti negativi

Gli aspetti negativi del progetto sono da legarsi principlamente alle tecnologie da utilizzare che sono poco familiari agli elementi del gruppo. La criticità maggiore è da riscontrarsi invece sulla fattibilità del progetto stesso che cerca una soluzione ad un problema piuttosto complesso che richiede grandi capacità di pensiero e di sviluppo.

# Gli altri progetti

## 3.1 NomeCapitolato

### 3.1.1 Valutazione Generale

-inserire valutazione-

### 3.1.2 Potenziali Criticità

-inserire criticità-

# Elenco delle tabelle