



Studio di Fattibilità

Gruppo SWEetBIT — Progetto SWEDesigner

Informazioni sul documento

Versione	1.0.0
Redazione	Salvatore Pilò
Verifica	Da inserire
Approvazione	Da inserire
Uso	Interno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo SWEetBITSWEetBIT

Descrizione

Questo documento descrive lo studio di fattibilità del capitolato analizzando i rischi e i benefici dello stesso in relazioni agli altri capitolati presenti.

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Scopo del Prodotto	3
1.3	Glossario	3
1.4	Riferimenti	3
2	Scelta del Capitolato C6	5
2.1	Descrizione del capitolato	5
2.2	Dominio applicativo	5
2.3	Dominio tecnologico	5
2.4	Criticità potenziali e costi	6
2.5	Analisi del mercato e benefici	6
2.6	Considerazioni e valutazioni finali	7
3	Altri capitolati	8
3.1	Capitolato C1 - An API Market Platform (APIM)	8
3.2	Capitolato C2 - Accoglienza tramite Assistente Virtuale(AtAVi)	8
3.3	Capitolato C3 - A Designer and Geo-localizer Web App for Organizational Plants(DeGeOP)	9
3.4	Capitolato C4 - Applicazione di lettura per dislessici(eBread)	9
3.5	Potenziati Criticità	9
3.6	Capitolato C5 - An interactive bubble provider(Monolith)	10

Versioni del documento

Versione	Data	Persone coinvolte	Descrizione
1.0.0	2017/03/02	Salvatore Pilò (Analista)	Creazione scheletro del documento, stesura introduzione e analisi capitolato

1 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 SWEt BIT.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di una *Web App_G* che fornisca all'utente un *UML_G Designer_G*, con il quale riuscire a disegnare correttamente diagrammi delle classi e descrivere il comportamento dei metodi interni alle stesse attraverso l'utilizzo di diagrammi delle attività. La *Web App_G* permetterà all'utente di generare codice Java dal diagramma disegnato ed eventualmente andare a ritoccarne il risultato al fine di ottenere un codice eseguibile, funzionante 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: 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_G* 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_G 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*

2 Scelta del Capitolato C6

2.1 Descrizione del capitolato

Il capitolato C6, proposto dall'azienda *Zucchetti S.p.a.*, propone lo sviluppo di una *Web App_G* costituita da un *designer_G* di *UML_G* che possa utilizzare sia gli schemi tipici del linguaggio, come ad esempio il diagramma delle classi, sia alcuni ibridi ideati appositamente per lo scopo. Dal diagramma prodotto sarà possibile generare automaticamente del codice *Java*. Le richieste principali del capitolato sono le seguenti:

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

2.2 Dominio applicativo

Il capitolato pone come obiettivo quello di creare uno strumento che possa automatizzare, nei limiti 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 del codice quanto più pulito possibile da errori umani, pertanto nasce l'esigenza di un tool in grado di automatizzare questo processo macchinoso, rendendo meno influente l'azione umana, ed i relativi errori, sul prodotto finale.

Nella pratica un tale sistema sarebbe impossibile da realizzare per via della mole di variabili in gioco, pertanto occorre provare a ridimensionare il problema ponendolo all'interno di un dominio specifico. In questo caso il dominio indicato dal proponente è quello dei giochi da tavolo, 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 proprio vantaggio.

2.3 Dominio tecnologico

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

- ***Server_G* 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:** Conoscenza abbastanza approfondita del linguaggio, necessaria per la generazione del codice automatico;
- **Diagrammi UML_G :** Conoscenza dei principali schemi utilizzati all'interno dello standard UML_G ;
- **Meteor:** Conoscenza basilare della piattaforma per agevolare la scrittura del lato $Client_G$ della $Web App_G$.

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 essere fonti di forti criticità sono le seguenti:

- **Diagrammi UML_G :** Nessun componente del gruppo ha mai avuto a che fare con la progettazione di diagrammi UML_G . Lo studio approfondito di tali strumenti è fondamentale per la realizzazione del progetto;
- **Java:** il gruppo possiede una conoscenza piuttosto generale del 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 $Server_G$, si rende pertanto necessaria una conoscenza generale per la scelta della tecnologia da adoperare da approfondire maggiormente in seguito;
- **Meteor:** nonostante i componenti 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

Attualmente sul mercato non sono disponibili strumenti di questo genere e che offrano di generare del codice in maniera automatizzata. I pochi esempi che possiamo ritrovare prevedono un sistema poco funzionale di *Drag-and-drop_G* che genera del codice non sempre ottimale. Oltre a questo si sente molto l'esigenza di un ambiente 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 crescita rapida 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, il 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;
- **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 principalmente alle tecnologie da utilizzare, che sono poco familiari agli elementi del gruppo. La criticità maggiore è da riscontrarsi invece sulla fattibilità del progetto stesso, il quale cerca una soluzione ad un problema piuttosto complesso che richiede grandi capacità di pensiero e di sviluppo.

3 Altri capitoli

3.1 Capitolato C1 - An API Market Platform (APIM)

3.1.1 Valutazione Generale

Il capitolato propone la creazione di una *Web App_G*, che consiste in un *API_G* market in grado di registrare, consultare ed effettuare operazioni di compravendita di *microservizi_G*. Il gruppo ha ritenuto il capitolato C1 fattibile e stimolante per la possibilità di interagire con una tecnologia di recente espansione, ovvero l'architettura a *microservizi_G*. Tuttavia non ha suscitato molto interesse la limitazione delle tecnologie necessarie, si è deciso di effettuare la propria scelta su altri capitoli.

3.1.2 Potenziali Criticità

- Difficoltà nel garantire la correttezza delle *API_G* registrate dagli utenti.

3.2 Capitolato C2 - Accoglienza tramite Assistente Virtuale (AtAVi)

3.2.1 Valutazione Generale

Il capitolato propone la creazione di una *Web App_G* che permetta, ad un ospite dell'ufficio dei proponenti, di interrogare un assistente virtuale per annunciare la propria presenza, in modo che l'applicativo lo accolga e comunichi l'arrivo a chi di dovere. Il gruppo ha ritenuto il capitolato C2 molto affascinante ed eccitante, soprattutto perché si affronta la tematica dell'*Intelligenza Artificiale_G*, una tematica fortemente attuale e che è destinata a diventare sempre più fondamentale in innumerevoli settori. Nonostante ciò, il gruppo ha optato su altri capitoli poiché l'inesperienza dei componenti su un argomento così complesso avrebbe potuto aumentare notevolmente la difficoltà del capitolato fuoriuscendo obbiettivi iniziali di quest'ultimo.

3.2.2 Potenziali Criticità

- Difficoltà nel creare un programma di *IA_G* efficiente;
- Scarse conoscenze riguardo gli *SDK_G* per assistenti virtuali, quindi difficoltà nell'effettuare paragoni ed analisi;
- Conoscenze basilari solamente da parte di alcuni membri del gruppo di NodeJS, con conseguente incremento del tempo per l'apprendimento di esso.

3.3 Capitolato C3 - A Designer and Geo-localizer Web App for Organizational Plants (DeGeOP)

3.3.1 Valutazione Generale

Il capitolato richiede la creazione di un interfaccia *Web App_G*, erogabile anche su dispositivi mobili, per inserire i processi produttivi delle aziende (macchinari, magazzini, fornitori, distributori) su mappa geografica e per disegnare i vari scenari di danno che possono colpire l'azienda. Questo capitolato non è stato ritenuto interessante dal gruppo sia dal punto di vista del dominio applicativo, sia delle tecnologie da utilizzare. Di conseguenza si è preferito scegliere altro.

3.3.2 Potenziali Criticità

- Difficoltà di definizione di tutti gli scenari di danno possibili.

3.4 Capitolato C4 - Applicazione di lettura per dislessici (eBread)

3.4.1 Valutazione Generale

L'obiettivo di questo capitolato è quello di realizzare un'applicazione in ambiente *Android_G* che agevoli la lettura alle persone affette da dislessia, grazie all'aiuto di tecnologie appropriate, fra cui la sintesi vocale. Il gruppo ha deciso di non approfondire questo capitolato perché, considerando l'alto numero di applicazioni appartenenti allo stesso dominio, con il tempo a disposizione sarebbe stato complicato ottenere innovazioni degne di nota. Si è quindi preferito puntare su capitoli più originali.

3.5 Potenziali Criticità

- Difficoltà di implementazione di un motore di sintesi vocale che sia sincronizzato con il testo;
- Difficoltà di implementazione di supporto multilingua;
- Scarsa conoscenza da parte del gruppo delle tecnologie da utilizzare.

3.6 Capitolato C5 - An interactive bubble provider (Monolith)

3.6.1 Valutazione Generale

Il capitolato prevede la creazione di un *framework_G* che permetta l'istanziamento delle cosiddette bolle per la piattaforma di *Web Chat_G* denominata Rocket.chat, dove per bolle si intendono delle funzionalità che possono venire aggiunte alla piattaforma senza nessuna nuova installazione. Il gruppo ha reputato questo capitolato poco interessante dato che ormai esistono numerose piattaforme di Web Chat affermate e note a milioni di utenti, quindi si sarebbe difficilmente arrivati ad una vera innovazione.

3.6.2 Potenziali Criticità

- Difficoltà di contatto con i proponenti, vista la locazione della loro sede.