

# ANALISI DEI REQUISITI

SWEg Group

# 1 Registro Modifiche

Modifica	Nome	Data	Ver.
Creazione Raw Documento	Gianluca Crivellaro	21/12/2016	0.0.1
Modifica Raw	Gianluca Crivellaro	22/12/2016	0.0.2
Aggiunti Requisiti	Sebastiano Marchesini	23/12/2016	0.0.3
Aggiunti Requisiti Estesi Concordati	Gianluca Crivellaro	27/12/2016	0.0.4
Stesura Documento (Introduzione)	Sebastiano Marchesini	28/12/2016	0.1.0
Creazione Grafici UML	Pietro Lonardi e Gianluca C.	29/12/2016	0.1.1
Stesura Documento (Descrizione Generale & Casi d'Uso)	Sebastiano Marchesini	30/12/2016	0.1.2
Creazione Grafici UML	Pietro Lonardi	02/01/2017	0.1.3
Verifica Documento & Grafici UML	Piergiorgio Danieli	03/01/2017	0.1.4

# Indice

<b>1</b>	<b>Registro Modifiche</b>	<b>1</b>
<b>2</b>	<b>Introduzione</b>	<b>3</b>
2.1	Scopo del Documento . . . . .	3
2.2	Scopo del Prodotto . . . . .	3
2.3	Glossario . . . . .	3
2.4	Riferimenti Normativi e Informativi . . . . .	3
2.4.1	Normativi . . . . .	3
2.4.2	Informativi . . . . .	4
<b>3</b>	<b>Descrizione Generale</b>	<b>5</b>
3.1	Obbiettivi del prodotto . . . . .	5
3.2	Funzioni del prodotto . . . . .	5
3.3	Caratteristiche degli utenti . . . . .	6
3.3.1	Tipologia di Utenti . . . . .	6
3.4	Piattaforma di esecuzione . . . . .	6
3.5	Vincoli generali . . . . .	6
<b>4</b>	<b>Caso d’uso</b>	<b>7</b>
4.1	Caso d’uso Generale: APIM . . . . .	7

## 2 Introduzione

### 2.1 Scopo del Documento

Tale documento ha lo scopo di studiare e modellare concettualmente il problema che si pone con APIM. Posizionando le componenti (o ambiti) a scopo di allocazione dei requisiti. Alcuni dei requisiti specificandoli con il diagramma dei casi d'uso.

Vi deve essere la certezza di non aver lasciato dimenticato nessuno tra i bisogni espliciti e i bisogni impliciti. Questo implica che non vi sia ambiguità tra i requisiti.

Bisogna sempre tener conto di portare al massimo possibile la granularità del problema, senza però confonderlo e renderlo impossibile da verificare. Questo per rendere il requisito decidibile.

E' infine bene tener presente otto semplici qualità di selezione dei requisiti:

- Non Ambigui
- Corretti
- Completi
- Verificabili
- Consistenti
- Modificabili
- Tracciabili
- Ordinati per Rilevanza

### 2.2 Scopo del Prodotto

L'obiettivo è creare un'infrastruttura che permetta la distribuzione digitale e la gestione dei diritti digitali di microservizi. Creati e importati da diversi utenti che possono interfacciarsi tra loro.

Viene usata per gestire e distribuire una vasta gamma di microservizi (alcuni esclusivi) e il loro relativo supporto. Tutte queste operazioni sono effettuate via Internet.

E' inoltre possibile il monitoraggio di ogni API grazie alle tecnologie fornite dal prodotto.

### 2.3 Glossario

Alla fine di evitare ambiguità e mantenere la consistenza il Glossario è un documento unico e consultabile separatamente.

Un glossario è una raccolta di termini di un ambito specifico e circoscritto. In questo caso per raccogliere termini desueti e specialistici inerenti al progetto.

### 2.4 Riferimenti

#### 2.4.1 Normativi

- **Norme di Progetto:** "Norme di Progetto v1.0.0".
- **Capitolato d'appalto C1:** API Market per microservizi  
[www.math.unipd.it/~tullio/IS-1/2016/Progetto/C1.pdf](http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C1.pdf).
- **Verbali:**

### 2.4.2 Informativi

- **Studio di Fattibilità:** "Studio di Fattibilità v.1.0.0".
- **IEEE 830-1998:** Recommended Practice for Software Requirements Specifications  
[https://en.wikipedia.org/wiki/Software\\_requirements\\_specification](https://en.wikipedia.org/wiki/Software_requirements_specification).

## 3 Descrizione Generale

### 3.1 Obbiettivi del prodotto

L'obiettivo primario del prodotto é di dare ad ogni utente la possibilità di registrare il proprio micro-servizio in una piattaforma dedicata. In questo modo è possibile la vendita (o condivisione) con gli altri utenti della comunità regolata da politiche di compravendita specifiche e flessibili a seconda dello scopo dell'API o del volere del tecnico.

L'obiettivo è quindi quello di incentivare la programmazione a microservizi e, oltre a spingere i gruppi più piccoli nel progettare per il mercato virtuale, pensare sempre di più a delle migliori architetture flessibili invece che veri e propri programmi. Si vuole abbandonare i vecchi programmi monolite per entrare in una realtà fatta di sistemi divisi in moduli, la nuova sfida progettuale sarà quindi unire i vari microservizi (o API) per costruire un prodotto completo.

### 3.2 Funzioni del prodotto

SWEg Group si impegna in particolar modo alle sottoscritte funzioni del prodotto :

1. **Registrare le API di un microservizio:** dando la possibilità di caricare un interfaccia e documentando la propria progettazione.
2. **Permetta di consultare le API:** con un sistema di ricerca designato e filtrato anche con dati tecnici . Anche se con minori funzionalità anche un utente non registrato alla piattaforma può vagliare le varie API. Per ogni api sarà inoltre possibile un consulto dei suoi dati tecnici.
3. **Permetta di associare diverse API key:** così da regolare le politiche di scambio dei microservizi. Le API key sono lo strumento principale di collegamento tra la API e il suo utilizzatore. Grazie a queste l'infrastruttura potrà regolare le scadenze , l'utilizzo e procedimento oltre ad avere un ID univoco per la monitorizzazione.
4. **Permetta di monitorare l'utilizzo delle API:** già accennato nei punti precedenti. Vogliamo che tale infrastruttura tenga conto di particolari dati tecnici di ogni API per renderle così misurabili in termini di efficacia ed efficienza. Oltre che a così avere un sistema automatizzato per il confronto tra i vari microservizi.
5. **Blocchi le chiamate di utenti in possesso di API key scadute e/o non valide:** è la sottolineatura di uno dei motivi di esistenza delle API key. Punto focale per la regolamentazione dello scambio è la possibilità di acquisto delle chiavi secondo tempo, mole di scambio di dati , eccetera. I dati tecnici per le policy di durata e validità saranno descritte in seguito, ma queste decideranno se è ancora attiva una chiave o meno.
6. **Permetta di visualizzare i dati tecnici d'uso delle singole API:** dopo aver monitorato ogni singola API è possibile fare la stima e produrre un elaborato tecnico dei valori di quest'ultima. E' compito dell'API Market rendere disponibile questa funzionalità. Da parte nostra vi sarà un vaglio tra le principali e caratteristiche di interesse da dover riportare. E' da parte nostra desiderabile anche la possibilità di poter visualizzare direttamente il confronto dei risultati delle caratteristiche per scegliere il microservizio migliore.
7. **Permettere di gestire una moneta virtuale per la compravendita delle API:** il metodo di acquisto principale è comunque la moneta reale. Che può trasformarsi automaticamente in moneta virtuale con un cambio di 1:1. E' possibile quindi tenere un conto personale flessibile per poter reinvestire o ritirare il contante virtuale.

8. **Permetta di confrontare i dati tecnici delle API tra loro:** già accennato in uno dei punti precedenti. Per una migliore visione e per scegliere il microservizio più adatto ai nostri scopi vorremo una sezione specifica di confronto dei dati tecnici. Sarà nostro compito grazie al monitoraggio avere un rapporto aggiornato e reale dell'andamento dell'API.
9. **Permetta una gestione social:** vogliamo che quindi anche gli utenti abbiano le loro statistiche per essere valutati. Desiderabile un programma di messaggistica interno per una comunicazione diretta e veloce. Tutto è quindi classificabile, la valutazione degli utenti delle API crea delle classifiche stimolando la concorrenza e il desiderio di popolare il market di più microservizi. Vi saranno quindi graduatorie per genere grazie alle personali esperienze e ai dati tecnici da favorire l'interazione tra gli attori della piattaforma.

### 3.3 Caratteristiche degli utenti

Il prodotto è soprattutto proiettato a degli utenti secondo noi esperti nelle tecnologie impiegate. Abbiamo all'unisono prestabilito con il proponente del capitolato che è compito dell'utente creare l'interfaccia adeguata a registrare il microservizio nel market. Questo ci assicura una minima competenza nella programmazione e quindi nell'uso di tecnologie informatiche.

Dimestichezza nell'utilizzo di browser che sia esso da smartphone o da notebook.

#### 3.3.1 Tipologia di Utenti

Saranno 3 i principali tipi di utenti che andranno a popolare i nostri casi d'uso a seconda del livello di autenticazione nella piattaforma:

- *Utente non Autenticato:* è un utente che può visitare i microservizi in forma anonime a senza registrazione. Sarà limitato nelle impostazioni di ricerca, non potrà accedere alla funzionalità sociali e di compravendita delle API.
- *Utente Autenticato:* dopo essersi registrato nel data base della piattaforma l'utente autenticato ha accesso completo alle funzionalità offerte al pubblico. Può cambiare le proprie referenze, accumulare monete virtuali e avere pieno accesso alla parte sociale.
- *Amministratore:* l'amministratore può modificare i profili di tutti gli utenti, modificare tutti i microservizi caricati e ha accesso a una pagina sulle statistiche del sito. Ha le funzionalità più estese e si è nominati personalmente.

### 3.4 Piattaforma di esecuzione

Il prodotto finale è fruibile da qualsiasi piattaforma che disponga di un browser per la navigazione web. Sarà però garantito il corretto e perfetto funzionamento solo con alcune versioni e particolari browser. La parte di back end invece sarà

### 3.5 Vincoli generali

Per utilizzare le funzionalità della piattaforma è obbligatorio avere una connessione internet.

## 4 Caso d'uso

L'analisi del capitolato, il dibattito tra gli Analisti e l'incontro con ItalianaSoftware ha portato alla creazione dei casi d'uso che seguono. Si è cercato anche di studiare piattaforme sociale e di scambio simili come *Steam*, *Google Play* e *GitHub*.

Le fonti su cui si ricavano quindi i casi d'uso sono sia impliciti, derivanti dallo studio del dominio, sia espliciti, come il capitolato d'appalto. Ogni caso d'uso ha un codice univoco gerarchico, nella forma:

UC[codice univoco del padre].[codice progressivo di livello]

Il codice progressivo può includere diversi livelli di gerarchia separati da un punto.

### 4.1 Caso d'uso Generale: APIM