Piano di qualifica

SWEg Group

04/01/2017

Modifica	Nome	Data	Ver.
Studio dei riferimenti	Crivellaro Gianluca	2/01/2017	0.0.1
Stesura introduzione	Piergiorgio Danieli	04/01/2017	0.1.0
Stesura del documento	Crivellaro Gianluca	04/01/2017	0.2.0

Indice

1 Introduzione

1.1 Scopo del documento

Lo scopo del *Piano di Qualifica* è di descrivere le strategie di verifica e validazione adottate dal gruppo SWEg al fine di perseguire obiettivi qualitativi per il progetto APIMarket.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di un applicazione web che gestisca e monitori microservizi, muniti di un'interfaccia Jolie. In particolare APIMarket deve permettere di vendere, acquistare e monitorare microservizi tramite la piattaforma.

1.3 Glossario

Per evitare il più possibile ambiguità legate al linguaggio, e per favorire la comprensione del documento i termini tecnici e gli acronimi che necessitano di descrizione saranno seguiti da una "g" in pedice e riportati nel documento *Glossario v1.0.0*.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0*;
- **Capitolato d'appalto C1:** APIM: An API$_g$ Market Platform
  http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/Capitolati.html

1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v1.0.0*;
- **Slide di Ingegneria del Software:**
  http://www.math.unipd.it/~tullio/IS-1/2016/;
- **SWEBOK - Version 3 (2004)**: capitolo 11 - Software Quality
  https://www.computer.org/web/swebok;
- **ISO$_g$/IEC$_g$ TR 15504:** Software process assessment:
  https://en.wikipedia.org/wiki/ISO/IEC_15504;
- **ISO$_g$/IEC$_g$ 9126: Product quality**:
  https://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Nielsen$_g$ e le sue 10 euristiche**:
  http://www.far.unito.it/usabilita/Cap5.htm

2 Definizione obiettivi di qualità

Il gruppo SWEg si impegna a rispettare il modello di standard definito in ISO$_g$/IEC$_g$0126, il quale ha le seguenti qualità:

2.1 Funzionalità

Il sistema prodotto deve garantire tutte le funzionalità indicate nel documento *"Analisi dei Requisiti v1.0.0"*. L'implementazione dei requisiti deve essere più completa ed economica possibile.

- **Misura:** viene usata come unità di misura la quantità di requisiti mappati in componenti del sistema create e funzionanti;
- **Metrica:** la sufficienza è stabilita nel soddisfacimento di tutti i requisiti obbligatori;
- **Strumenti:** perché questa qualità sia soddisfatta il sistema deve superare tutti i test previsti.

2.2 Affidabilità

Il sistema si deve dimostrare robusto e di facile ripristino in caso di errori.

- **Misura:** l'unità di misura sarà la quantità di esecuzioni del sistema andati a buon fine;
- **Metrica:** le esecuzioni dovranno soddisfare la più ampia gamma di funzionalità previste;
- **Strumenti:** da definire.

2.3 Usabilità

Il sistema prodotto deve risultare di semplice utilizzo per l'utente. Questo sistema deve allo stesso tempo soddisfare tutte le necessità dell'utente.

- **Misura:** i dieci punti dell'euristica di Nielsen_g;
- **Metrica:** la sufficienza è stabilita dal soddisfacimento del numero minimo di punti;
- **Strumenti:** perché questa qualità sia soddisfatta il sistema deve superare tutti i test previsti.

2.4 Efficienza

Il sistema deve fornire tutte le funzionalità nel più breve tempo possibile, con il minimo utilizzo di risorse.

- **Misura:** tempi di latenza per ottenere una risposta dall'applicazione web in condizioni normali o in caso di sovraccarico della rete;
- **Metrica:**la sufficienza viene stimata come un tempo di latenza minore del 20% su rete via cavo o Wi-Fi e del 30% su reti cellulari rispetto alla media dei tempi di latenza del server web;
- **Strumenti:** si veda il documento *"Norme di Progetto v1.0.0"*.

2.5 Manutenibilità

Il sistema prodotto deve essere comprensibile ed estensibile in modo facile e verificabile.

- **Misura:** l'unità di misura utilizzata saranno le metriche sul codice descritte nella sezione Misure e metriche;
- **Metrica:** le metriche che verranno rispettate sono descritte nella sezione Misure e metriche;
- **Strumenti:** si veda il documento *"Norme di Progetto v1.0.0"*.

2.6 Portabilità

Il sistema deve essere più portabile possibile. Il sito deve essere visitabile da più browser possibili. Il sito deve essere sviluppato con sistemi che abbiano le varie componenti tecnologiche di tipo standard. Con il termine "standard" si intente che i contenuti devono poter essere utilizzati su più sistemi operativi possibile e su più ambienti di lavoro possibile.

- **Misura:** il front end deve aderire agli standard W3C;
- **Metrica:** se il software avrà la sufficienza in tutte le metriche descritte nella sezione Misure e metriche, allora il sito avrà le caratteristiche di portabilità descritte;
- **Strumenti:** si veda il documento *"Norme di Progetto v1.0.0"*.

2.7 Altre qualità

Saranno inoltre importanti per il prodotto le seguenti qualità:

- **Semplicità**: si ottiene tramite la realizzazione del prodotto nel modo più semplice, senza tralasciare la qualità;
- **incapsulamento:** applicare le tecniche di incapsulamento per aumentare la manutenibilità e la possibilità di riuso del codice. L'interazione sarà possibile tramite interfacce;
- **Coesione:** riguarda le funzionalità che collaborano tra di loro per ottenere uno stesso obiettivo. Devono risiedere nello stesso componente, e come scopo hanno quello di ridurre l'indice di dipendenza, favorire la semplicità e la manutenibilità.

3 Visione generale delle strategie di verifica

3.1 Procedure di controllo di qualità di processo

La qualità dei processi verrà garantita dall'applicazione del principio PDCA_{g}. Grazie a questo principio sarà possibile garantire non solo il controllo e la correttezza, ma anche il miglioramento costante della qualità di tutti i processi. Come conseguenza diretta si otterrà il miglioramento del prodotto.

Per avere controllo dei processi, e di conseguenza qualità, è necessario che:

- vi sia controllo sull'operato dei membri del team e sui processi;
- i processi siano pianificati in modo dettagliato;
- le risorse siano ripartite in modo chiaro.

L'attuazione di questi punti è descritta nel documento *"Piano di progetto v1.0.0"*. Inoltre l'analisi contante della qualità del prodotto permette di monitorare in modo indiretto la qualità dei processi: se il prodotto è di bassa qualità sicuramente il processo è migliorabile.

Per valutare la qualità di un processo E' fondamentale che questa sia quantificata, le metriche per fare ciò sono descritte nella sezione....

3.2 Procedure di controllo di qualità di prodotto

Il controllo di qualità dei prodotti verrà garantito dai seguenti processi:

- **Software Quality Assurance** (SQA_g): assicura che i processi siano appropriati per il progetto, che siano correttamente implementati. Prevede l'attuazione di tecniche di analisi statica e dinamica, descritte nel documento *"Norme di Progetto v1.0.0"*;
- **verifica:** è il processo che controlla la coerenza e la correttezza dei prodotti dei processi. La verifica verrà eseguita costantemente durante tutta la durata del progetto.
- **validazione:** ovvero la conferma che il sistema soddisfi i requisiti e sia conforme alle attese.

3.3 Organizzazione

Il processo di verifica inizia nel momento in cui la versione del prodotto di un processo cambia. Il diario delle modifiche aiuta a monitorare solo le sezioni che sono state modificate, ottimizzando così i tempi di verifica.

Viene verificata sia la qualità del processo che del prodotto da esso compiuto.

Ogni fase del progetto, descritta nel documento *"Piano di progetto v1.0.0"* produce un prodotto diverso, quindi necessita di diverse attività di verifica:

- **Analisi:** in questa fase si verifica che i processi e la documentazione prodotta rispetti il le norme definite nel documento *"Norme di Progetto v1.0.0"*
- **Analisi di Dettaglio:**...
- **Analisi Architetturale:**...

3.4 Pianificazione strategica e temporale

E' essenziale che l'attività di verifica sia sistematica e organizzata, al fine di evitare la propagazione di errori e ottimizzare i tempi di sviluppo, per rispettare le scadenze elencate nel documento *Piano di Progetto v1.0.0*.

Ogni fase di redazione di documenti e di codifica deve essere preceduta da una fase di studio preliminare, al fine di ridurre la possibilità di errori di natura concettuale o tecnica, e favorendo l'attività dei verificatori.

In seguito vengono riportate le scadenze fissate:

- **Revisioni formali:**
  - **Revisione dei Requisiti:** 24/01/2017;
  - **Revisione di Accettazione:** 27/06/2017
- **Revisioni di progresso:**
  - **Revisione di Progettazione:** 13/03/2017
  - **Revisione di Qualifica:** 15/05/2017

3.5 Responsabilità

Al fine di garantire un processo di verifica efficace ed efficiente vengono attribuite delle responsabilità all'interno del gruppo di progetto.

I ruoli che intervengono nel processo di verifica sono il *Project Manager* e il *Verificatore*. La suddivione dei compiti è descritta nel documento *"Norme di Progetto v1.0.0"*.

3.6 Risorse

Per realizzare il prodotto sono necessarie le seguenti risorse:

- **Risorse umane:** vengono descritte dettagliatamente nel documento *"Piano di Progetto v1.0.0"*.
  - Project Manager;
  - Amministratore;
  - Analista;
  - Progettista;
  - Programmatore;
  - Verificatore.
- **Risorse software:** sono necessari strumenti software per:
  - scrivere la documentazione in formato LaTeX$_g$;
  - creare diagrammi *UML*$_g$;
  - automatizzare le verifiche;
  - sviluppare nei linguaggi di programmazione scelti;
  - analizzare il codice scritto;
  - gestione dei test sul codice.
- **Risorse hardware:** sono necessari computer con caratteristiche descritti nel documento *"Norme di Progetto v1.0.0"*.

3.7 Tecniche di analisi

3.7.1 Analisi statica

L'analisi statica è l'analisi eseguita senza una effettiva esecuzione dei programmi. Viene eseguita quindi sia al codice che ai documenti durante tutta la fase di sviluppo.

Può essere applicata nei seguenti modi:

- **Walkthrough:** é una lettura del documento da verificare cercando errori senza un'idea precisa di cosa si sta cercando. E' utile nelle prime fasi di sviluppo per stilare una lista degli errori più frequenti da condividere con il gruppo,
- **Inspection:** è una ricerca mirata degli errori che sono stati evidenziati nella lista di controllo, che può essere espansa qualora se ne palesi la necessità.

3.7.2 Analisi dinamica

L'analisi dinamica viene eseguita sulle componenti software, testando il comportamento dell'applicativo. I test, per essere attendibili, devono essere ripetibili nello stesso ambiente e con stessi input.

I tipi di test che verranno eseguiti saranno:

- **Test di unità:** consiste nel controllo dei moduli che costituiscono l'applicativo;
- **Test di integrazione:** consiste nella verifica che i moduli, una volta collegati tra loro, diano il risultato previsto. Verifica inoltre il corretto funzionamento di librerie$_g$ e framework$_g$ esterni;

- **Test di sistema:** consiste nella validazione del prodotto software, una volta giunto ad uno stadio definitivo.
- **Test di regressione:** consiste nell'eseguire a cascata tutti i test collegati ad una componente appena modificata;
- **Test di accettazione:** consiste nel collaudo del software con il proponente prima del rilascio.

3.8 Misure e metriche

Qui di seguito vengono elencate le misure e le metriche che verranno adottate nel processo di verifica. Il processo di verifica deve essere verificabile e quantificato secondo metriche stabilite a priori.

Vi possono essere due tipologie di range nell'accettazione dei risultati di un processo di verifica:

- **Accettazione:** è il livello minimo che il prodotto deve raggiungere per superare il test;
- **Ottimale:** è il livello in cui dovrebbe posizionarsi la misurazione. In caso questo valore non venga raggiunto sono necessarie verifiche aggiuntive.

3.8.1 Metriche per i processi

3.8.1.1 Schedule Variance (SV)

Indica se si è in linea, in anticipo o in ritardo rispetto alla schedulazione delle attività di progetto pianificate nella baseline. È un indicatore di efficacia soprattutto nei confronti del Cliente. Se SV > 0 significa che il progetto sta producendo con maggior velocità a quanto pianificato, viceversa se negativo.

Parametri utilizzati:

- **Range di accettazione:** $\geq$ -(preventivo fase*5%);
- **Range ottimale:** ≥ 0.

3.8.1.2 Budget Variance (BV)

Indica se alla data corrente si è speso di più o di meno rispetto a quanto previsto a budget alla data corrente. È un indicatore che ha un valore unicamente contabile e finanziario. Se BV > 0 significa che il progetto sta spendendo il proprio budget con minor velocità di quanto pianificato, viceversa se negativo.

Parametri utilizzati:

- **Range di accettazione:**$\geq$-(preventivo fase*10%);
- **Range ottimale:**≥ 0.

3.8.2 Metriche per i documenti

3.8.2.1 Gulpease

L'indice Gulpease è un indice, sviluppato per valutare la lingua italiana, che valuta la complessità e la leggibilità del testo. Rispetto ad altri ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. L'indice di Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

L'indice è calcolato con la seguente formula:

$$89 + \frac{300 * (numero\ delle\ frasi) - 10 * (numero\ delle\ lettere)}{(numero\ delle\ parole)}$$

I risultati sono compresi tra 0 e 100, dove il valore "100" indica la leggibilità più alta e "0" la leggibilità più bassa. In generale risulta che testi con un indice:

- inferiore a 80 sono difficili da leggere per chi ha la licenza elementare;
- inferiore a 60 sono difficili da leggere per chi ha la licenza media;
- inferiore a 40 sono difficili da leggere per chi ha un diploma superiore.

Va notato che questo indice non considera che il testo sia comprensibile o meno, in effetti anche una frase con parole casuali potrebbe avere un ottimo indice Gulpease. I documenti perciò vanno valutati da un essere umano per rendere le frasi semplici ma comprensibili.

Parametri utilizzati:

- **Range di accettazione:** [40 - 100];
- **Range ottimale:** [50 - 100].

3.8.2.2 Metriche per il software

3.8.2.3 Complessità ciclomatica

La complessità ciclomatica è una metrica software utilizzata per misurare la complessità di un programma. Misura direttamente il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso.

La complessità ciclomatica è calcolata utilizzando il grafo di controllo di flusso del programma: i nodi del grafo corrispondono a gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo di istruzioni può essere eseguito immediatamente dopo il primo gruppo.

Alti valori di questo indice indicano che il codice è poco manutenibile e difficile da testare, mentre valori bassi potrebbero indicare una scarsa efficienza nei metodi.

Parametri utilizzati:

- **Range di accettazione:** [1 - 15];
- **Range ottimale:** [1 - 10].

3.8.2.4 Rapporto linee di commeneto su linee di codice

Affinché il codice sia più facilmente comprensibile e manutenibile è necessario che sia adeguatamente commentato.

Parametri utilizzati:

- **Range di accettazione:** [>0.2];
- **Range ottimale:** [>0.3].

3.8.2.5 Numero di livelli di annidamento

Indica il livello di annidamento delle strutture di controllo. Un numero troppo elevato può comportare difficoltà nella verifica e nella manutenibilità del codice.

Parametri utilizzati:

- **Range di accettazione:[0-6]**
- **Range ottimale:[0-4]**

3.8.2.6 Numero di campi dati per classe

Una classe con troppi campi dati potrebbe essere poco specializzata e quindi mal progettata. Una classe poco specializzata inoltre è anche poco manutenibile.

Parametri utilizzati:

- **Range di accettazione:[0-15]**
- **Range ottimale:[1-8]**

3.8.2.7 Accoppiamento

L'indice di accoppiamento è il risultato di due indici:

- **Accoppiamento afferente:** è il numero di classi esterne ad un package che dipendono da classi interne ad esso. Un valore troppo elevato indica un altro grado di dipendenza, che potrebbe portare a criticità nel caso di modifiche.
- **Accoppiamento efferente:** è il numero di classi interne al package dipendenti da classi esterne. Un valore troppo alto è sintomo di scarsa progettazione.

Il calcolo dell'indice di stabilità, ottenuto unendo Accoppiamento afferente e Accoppiamento efferente avviene secondo la seguente formula:

$$I = \frac{Ce}{Ca + Ce}$$

3.8.2.8 Validazione W3C

Una pagina web per essere accessibile ed usabile deve essere valida secondo le norme del W3C. L'applicativo web deve superare tutti i test del validatore automatico W3C per quanto riguarda gli errori gravi, deve inoltre evitare quanti più warning possibili.

Parametri utilizzati:

- **Range di accettazione:[0-5]**
- **Range ottimale:[0-0]**

3.8.2.9 Euristica di Nielsen

Le euristiche di Nielsen sono 10 euristiche studiate per controllare in modo metodico l'usabilità di un sito web, derivano dall'applicazione di tecniche di analisi fattoriale su 249 problemi di usabilità. Sono semplici da applicare e mostrano benefici nelle fasi iniziali di un progetto, ma sono informali e non sono automatizzabili in quanto richiedono un giudizio umano. Essendo parametri non completamente oggettivi, può capitare che alcune pagine siano usabili anche non rispettando tutte le euristiche, si è quindi posto un limite minimo di 6 euristiche rispettate.

Parametri utilizzati:

- **Range di accettazione: 6/10**
- **Range ottimale: 10/10**

4 Gestione amministrativa della revisione

4.1 Comunicazione e risoluzione delle anomalie

Quando un verificatore trovi un'anomalia dovrà aprire un nuovo ticket secondo le modalità ripostate nel documento *"Norme di Progetto v1.0.0"*. Un'anomalia corrisponde a:

- Un errore ortografico;
- Una violazione delle norme tipografiche definite nel documento *"Norme di Progetto v1.0.0"*;
- Un'incongruenza el prodotto rispetto a determinate funzionalità stabilite nel documento *"Analisi dei Requisiti v1.0.0"*;
- Un'incongruenza del codice con il design prodotto;
- L'uscita dal range di accettazione degli indici di misurazione.

5 Resoconto delle attività di verifica

Nel periodo antecedente la consegna di ogni revisione i documenti ed i processi vengono verificati. I documenti vengono verificati tramite la procedura del walkthrough; in seguito sono state avviate le procedure per la notifica descritta nel documento *"Norme di Progetto v1.0.0"*.

Una volta individuati gli errori, vengono trattati nel seguente modo:

- Correzione degli errori individuati;
- Inserimento nella lista di controllo degli errori comuni;
- Applicazione del ciclo PDCA.

Successivamente viene applicato il metodo inspection ai diagrammi dei casi d'uso.