

24-05-2015



Piano di Qualifica

Informazioni sul documento

Nome Documento	Piano di Qualifica
Versione	1.1.0
Stato	<i>Formale</i>
Uso	<i>Esterno</i>
Data Creazione	20-03-2015
Data Ultima Modifica	24-05-2015
Redazione	Fossa Manuel, Petrucci Mauro
Approvazione	Tollot Pietro
Verifica	Gabelli Pietro
Lista distribuzione	<i>LateButSafe</i> Prof. Tullio Vardanega Prof. Riccardo Cardin Proponente Zucchetti S.p.a.

Registro delle modifiche

Tab 1: Versionamento del documento

Versione	Autore	Data	Descrizione
2.0.0	Petrucci Mauro	24-05-2015	Approvazione documento
1.7.0	Petrucci Mauro	24-05-2015	Apportate correzioni segnalate dal verificatore Gabelli Pietro
1.4.0	Gabelli Pietro	22-05-2015	Verifica documento
1.1.0	Venturelli Giovanni	04-05-2015	Inizio primo incremento: resi maggiormente coerenti i contenuti
1.0.0	Tollot Pietro	13-04-2015	Approvazione del documento
0.7.0	Petrucci Mauro	08-04-2015	Apportate le modifiche segnalate dal verificatore Fossa Manuel
0.3.0	Petrucci Mauro	25-03-2015	Aggiunta dei contenuti
0.2.0	Fossa Manuel	24-03-2015	Aggiunta dei contenuti
0.1.0	Busetto Matteo	20-03-2015	Stesura dello scheletro del documento



pre-RR

Tab 2: Storico ruoli pre-RR

Tab 3: Storico ruoli RR \rightarrow RP

Indice

Elenco delle figure

1	Rappresentazione del modello ISO/IEC 9126:2001	10
2	Schema PDCA	12

Elenco delle tabelle

1	Versionamento del documento	1
2	Storico ruoli pre-RR	2
3	Storico ruoli RR -> RP	2
4	Descrizione dei test di sistema per i Requisiti Funzionali	20
5	Descrizione dei test di sistema per i Requisiti di Qualità e Vincoli	28
6	Esiti verifica documenti, Analisi	29

Sommario

Il presente documento contiene le norme e le convenzioni che il gruppo LateButSafe intende adottare durante l'intero ciclo di vita del prodotto software Premi.

1 Introduzione

1.1 Scopo del documento

Il Piano di Qualifica ha lo scopo di descrivere le strategie che il gruppo di lavoro ha deciso di adottare per perseguire obiettivi qualitativi da applicare al proprio prodotto. Per ottenere tali obiettivi è necessario un processo_g di verifica continua sulle attività svolte; questo consentirà di rilevare e correggere anomalie e incongruenze in modo tempestivo e senza spreco di risorse_g.

1.2 Scopo del Prodotto

Lo scopo del progetto_g è la realizzazione un software_g per la creazione ed esecuzione di presentazioni multimediali favorendo l'uso di tecniche di storytelling e visualizzazione non lineare dei contenuti.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici e di dominio, gli acronimi e le parole che necessitano di essere chiarite sono riportate nel documento [Glossario_v.2.0.0.pdf](#). Ogni occorrenza di vocaboli presenti nel Glossario è marcata da una “g” minuscola in pedice.

1.4 Riferimenti

1.4.1 Normativi

- Norme di Progetto_g: [NormeDiProgetto_v.2.0.0.pdf](#);
- Capitolato d'appalto C4: Premi: Software_g di presentazione “better than Prezi” <http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf>.

1.4.2 Informativi

- Standard ISO_g /IEC 9126: Product quality
http://en.wikipedia.org/wiki/ISO/IEC_9126;



vengono considerate la sua tolleranza agli errori, la capacità di evitare fallimenti nell'esecuzione a seguito di malfunzionamenti (detta maturità) e la recuperabilità dei dati e delle prestazioni nell'eventualità di un malfunzionamento inevitabile. Il prodotto può considerarsi affidabile se il numero di esecuzioni andate a buon fine è sufficientemente grande rispetto al numero di esecuzioni totali.

2.2.3 Efficienza

È un requisito_g non funzionale che indica il rapporto tra le prestazioni e le risorse_g disponibili. Si valuta se il software_g utilizza al meglio le risorse_g a sua disposizione per fornire le funzionalità richieste, considerando il suo comportamento rispetto al tempo, ossia la velocità di risposta e d'elaborazione in determinate condizioni, che rispetto all'uso delle risorse_g, data dalla capacità d'utilizzarne una quantità adeguata ad eseguire le funzioni_g richieste.

Un modo per valutare l'efficienza di un software_g è calcolarne i tempi di attesa in seguito all'esecuzione di un comando, tuttavia, nel caso del prodotto Premi l'efficienza è limitata anche dallo stato della rete e dall'utilizzo di componenti grafiche quali video o immagini; per questo motivo il gruppo non può garantire tempi di risposta brevi per ogni azione compiuta dall'utente, ma si impegna a non appesantire ulteriormente tali componenti.

2.2.4 Usabilità

È un requisito_g non funzionale che indica la capacità del software_g di essere compreso, appreso ed usato con soddisfazione dall'utente.

Per far ciò il prodotto deve soddisfare condizioni di comprensibilità, apprendibilità ed operabilità; deve inoltre avere una certa attrattiva nei confronti dell'utente allo scopo di rendergliene piacevole l'utilizzo. Questa caratteristica non è facilmente misurabile in quanto non esistono metriche_g per quantificarla, perciò si farà affidamento alle linee guida del material design fornite da Google, dato l'alto tasso di adozione rispetto ad altre linee guida.

2.2.5 Manutenibilità

È un requisito_g non funzionale che indica la capacità del software_g di essere corretto, migliorato o adattato con impegno contenuto; a tale scopo esso deve essere facilmente analizzabile e modificabile, deve garantire stabilità a seguito di modifiche e la testabilità di tali modifiche.

Per misurare questa caratteristica esistono una serie di metriche_g descritte nella sezione 4.6.

2.2.6 Portabilità

È un requisito_g non funzionale che indica la capacità del software_g di adattarsi al cambio di dispositivo e sistema operativo, limitando la necessità di apportare cambiamenti.

Per soddisfare questa caratteristica, come espresso dal capitolato, è necessario che il software_g funzioni_g sia su computer (indipendentemente dal loro sistema operativo) e su dispositivi mobile_g Android_g, iOS e Windows_g Phone.

2.3 Procedure di controllo di qualità di processo

Per applicare il modello SPICE_g si utilizzerà il ciclo di Deming. Il ciclo di Deming è un sistema iterativo per il miglioramento continuo della qualità dei processi_g e dei prodotti da essi risultanti,



- **Numero linee di codice:** rappresenta il numero di linee di codice_g all'interno di un blocco. Un indice elevato non rappresenta necessariamente un cattivo codice_g ma suggerisce la possibilità di estrarre metodi contenenti gruppi di istruzioni correlate, aumentando il livello di astrazione;
- **Halstead:** la metrica di Halstead non è solamente un indice di complessità, ma identifica le proprietà misurabili del software e le relative relazioni. Si basa sull'osservazione che una metrica dovrebbe valutare l'implementazione di un algoritmo in linguaggi differenti ed essere indipendente dall'esecuzione su una specifica piattaforma.

Calcolo:

Prima di tutto bisogna ricavare, dal codice sorgente, i seguenti valori:

- $n1$ = numero distinti operatori;
- $n2$ = numero distinti operandi;
- $N1$ = numero totale operatori;
- $N2$ = numero totale operandi.

Successivamente possono essere calcolati i seguenti valori:

- **Program length:**

$$N = N1 + N2$$

- **Program vocabulary:**

$$n = n1 + n2$$

- **Volume:** il volume descrive la dimensione dell'implementazione di un algoritmo e si basa sul numero di operazioni eseguite e sugli operandi di una funzione. Il volume di una function senza parametri composta da una sola linea è 20, mentre un indice superiore a 1000 indica che probabilmente la funzione esegue troppe operazioni.

$$V = N \times \log_2(n)$$

Parametri utilizzati

- * Range-accettazione: [20-1500];
- * Range-ottimale: [20-1000];

- **Indice di manutenibilità:** Questa metrica è una scala logaritmica con valore massimo 171. Rappresenta quanto manutenibile è il codice_g, ossia quanto facile è da supportare e migliorare.

L'indice di manutenibilità è calcolato tramite una fattorizzazione di altre metriche come Linee di Codice(LOC), Complessità Ciclomatica(CC), volume di Halstead(VH) e percentuale di commenti(COM).

Un elevato valore indica un'ottima manutenibilità, bassi valori al contrario indicheranno una difficoltà nella fase di manutenzione e incremento del codice:

$$M = 171 - 5.2 \ln(HV) - 0.23(CC) - 16.2 \ln(LOC) + 50.0 \sin(\sqrt{2.46 * COM})$$

Tab 5: Descrizione dei test di sistema per i Requisiti di Qualità e Vincoli

