

13-04-2015



Piano di Qualifica

Informazioni sul documento

Nome Documento	Piano di Qualifica
Versione	1.0
Stato	<i>Formale</i>
Uso	<i>Esterno</i>
Data Creazione	20-03-2015
Data Ultima Modifica	13-04-2015
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.



Tab 1: Versionamento del documento

Versione	Autore	Data	Descrizione
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

Indice

1	Introduzione	6
1.1	Scopo del documento	6
1.2	Scopo del Prodotto	6
1.3	Glossario	6
1.4	Riferimenti	6
1.4.1	Normativi	6
1.4.2	Informativi	6
2	Obiettivi di qualità	8
2.1	Qualità di processo	8
2.2	Qualità di prodotto	8
2.2.1	Funzionalità	8
2.2.2	Affidabilità	8
2.2.3	Efficienza	9
2.2.4	Usabilità	9
2.2.5	Manutenibilità	9
2.2.6	Portabilità	9
2.3	Procedure di controllo di qualità di processo	9
3	Visione generale delle strategie di verifica	11
3.1	Organizzazione	11
3.2	Pianificazione strategica e temporale	11
3.3	Responsabilità	12
4	Risorse	12
4.1	Risorse necessarie:	12
4.1.1	Risorse umane	12
4.1.2	Risorse Hardware	13
4.1.3	Risorse software	13
4.2	Risorse disponibili	13
4.2.1	Risorse software	13
5	Strumenti,tecniche e metodi	14
5.1	Strumenti	14
5.2	Tecniche	14
5.3	Metodi	15
5.4	Metriche	16
5.4.1	Metriche per il codice	16
5.4.2	Metriche per i documenti	17
6	Gestione amministrativa della revisione	18
6.1	Comunicazione e risoluzione di anomalie	18

7	Resoconto delle attività di verifica	19
7.1	Riassunto delle attività di verifica	19
7.1.1	Revisione dei Requisiti	19
7.1.2	Documenti	19
8	Verifica dei Requisiti	20
8.1	Verifica dei Requisiti Funzionali	20
8.2	Verifica dei Requisiti di Qualità e Vincoli	27



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.



1.1 Scopo del documento

1.2 Scopo del Prodotto

1.3 Glossario

1.4 Riferimenti

1.4.1 Normativi

- ### 1.4.2 Informativi

- Università degli studi di Padova - 2014/2015

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

2 Obiettivi di qualità

2.1 Qualità di processo

Al fine di garantire la qualità del prodotto in ogni fase di realizzazione, si deve garantire la qualità dei processi che lo definiscono; per questo motivo si è deciso di utilizzare lo standard ISO/IEC 15504 denominato SPICE, che rende disponibili strumenti adatti a valutarli.

2.2 Qualità di prodotto

Per garantire la qualità del prodotto si è deciso di seguire le indicazioni fornite dallo standard ISO/IEC 9126:2001 sostituito dal successivo ISO/IEC 2510:2011. Questo documento fornisce un modello per valutare la qualità esterna (nell'ambiente di utilizzo) ed interna (indipendente dall'ambiente) di un software, individuando sei caratteristiche principali atte a rendere il prodotto qualitativamente accettabile.

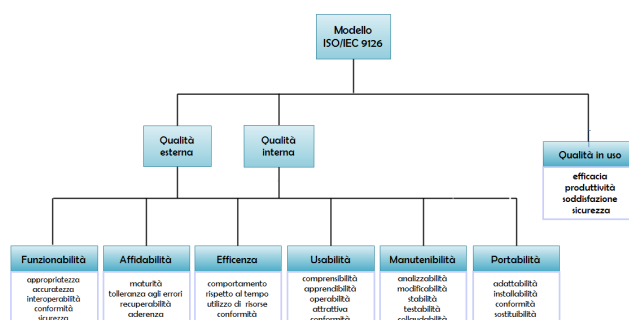


Fig 1: Rappresentazione del modello ISO/IEC 9126:2001

2.2.1 Funzionalità

È un requisito funzionale che indica la capacità del software di soddisfare le esigenze espresse dal capitolato ed individuate durante l'analisi dei requisiti. Per valutare questa caratteristica si considerano l'appropriatezza e l'accuratezza delle funzioni offerte, l'interoperabilità del prodotto rispetto ai diversi sistemi e la sicurezza offerta per la protezione dei dati.

Si sarà ottenuto un buon risultato in questo settore quando il software avrà superato in maniera positiva tutti i test e assicurerà copertura a tutti i requisiti obbligatori.

2.2.2 Affidabilità

È un requisito non funzionale che indica la capacità del software di svolgere correttamente il suo compito, mantenendo delle buone prestazioni anche al variare dell'ambiente nel tempo; 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 non funzionale che indica il rapporto tra le prestazioni e le risorse disponibili. Si valuta se il software utilizza al meglio le risorse 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, data dalla capacità d'utilizzarne una quantità adeguata ad eseguire le funzioni richieste.

Un modo per valutare l'efficienza di un software 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 non funzionale che indica la capacità del software 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 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 non funzionale che indica la capacità del software di essere corretto, migliorato o adattato con un impegno contenuto; a tale scopo esso deve essere facilmente analizzabile e modificabile, deve garantire stabilità a seguito di modifiche e testabilità di tali modifiche.

Per misurare questa caratteristica esistono una serie di metriche descritte nella sezione 5.4.

2.2.6 Portabilità

È un requisito non funzionale che indica la capacità del software 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 funzioni sia su computer (indipendentemente dal loro sistema operativo) e su dispositivi mobile Android, iOS e Windows Phone.

2.3 Procedure di controllo di qualità di processo

Per applicare il modello SPICE si utilizzerà il ciclo di Deming. Il ciclo di Deming è un sistema iterativo per il miglioramento continuo della qualità dei processi e dei prodotti da essi risultanti che permette di riconoscere lo stato di avanzamento di un progetto, fornendo un metodo di lavoro logico e sistematico.

È chiamato anche ciclo PDCA, in quanto è definito dall'iterazione delle quattro fasi:

- **Plan:** si stabiliscono obiettivi e processi necessari ad ottenere risultati conformi agli obiettivi attesi;



- **Do:** si implementa il piano, si esegue il processo e si realizza il prodotto. Si raccolgono dati da analizzare nei passi successivi;
- **Check:** si studiano i risultati ottenuti tramite la raccolta dei dati nella fase Do e si paragonano con i risultati attesi (gli obiettivi stabiliti nella fase Plan), per verificare la presenza di incongruenze. Si evidenziano le differenze nell'implementazione rispetto al piano;
- **Act:** se la fase di Check evidenzia che gli obiettivi fissati nel Plan e implementati nel Do rappresentano un miglioramento rispetto alla baseline precedente, si stabilisce una nuova baseline; in caso contrario la baseline non cambia. In entrambi i casi se la fase di Check ha evidenziato differenze rispetto alle aspettative, sarà necessario svolgere nuovamente il ciclo di PDCA.



3 Visione generale delle strategie di verifica

3.1 Organizzazione

Ogni qualvolta avvenga un cambiamento sostanziale nello sviluppo del prodotto, si istanzierà il processo di verifica.

Nello specifico durante ogni fase (Analisi, Progettazione, Realizzazione e Validazione) saranno applicate le tecniche di verifica qui descritte nei seguenti casi:

- Conclusione della prima redazione di un documento;
- Conclusione della prima redazione di un file di codice;
- Conclusione della modifica sostanziale di un documento: quando il versionamento passa da `.x.y.z` a `.x.y+1.0` oppure a `.x+1.0.0`. Si veda per approfondimento il paragrafo relativo al versionamento nel documento xxxxxx;
- Conclusione della modifica sostanziale di un file di codice, quando cioè il versionamento passa da `.x.y.z` a `.x.y +1.0` oppure a `.x+1.0.0`. Si veda per approfondimento il paragrafo relativo al versionamento nel documento xxxxxx.

L'obiettivo delle attività di verifica è quello di trovare e rimuovere i problemi presenti. Un problema può verificarsi a vari livelli, e per ogni livello assume un nome diverso:

- Fault (difetto): è l'origine del problema, ciò che fa scaturire il malfunzionamento;
- Error (errore): è lo stato per cui il software si trova in un punto sbagliato del flusso di esecuzione o con valori sbagliati rispetto a quanto previsto dalla specifica;
- Failure (fallimento, guasto): è un comportamento difforme dalla specifica, cioè la manifestazione dell'errore all'utente del software.

Esiste una relazione di causa-effetto fra questi tre termini:

$$DIFETTO \longrightarrow ERRORE \longrightarrow FALLIMENTO$$

Non sempre un errore dà origine ad un fallimento: ad esempio potrebbero esserci alcune variabili che si trovano in stato erraneo ma non vengono lette, o non viene percorso il ramo di codice che le contiene.

E' necessario prestare particolare attenzione a questo tipo di errori (detti anche quiescenti), avvalendosi anche di strumenti per il rilevamento dei bug.

3.2 Pianificazione strategica e temporale

Ai fini di rendere sistematica l'attività di verifica, per poter rispettare le scadenze fissate nel Piano di Progetto ed evitare la propagazione di errori all'interno dei documenti o di file di codice prima della loro verifica, la loro redazione sarà anticipata da una fase di studio preliminare.

Questa fase permetterà di ridurre la necessità di intervenire con grossi interventi a posteriori, quando la correzione di imprecisioni concettuali e tecniche potrebbe risultare particolarmente



gravosa.

Come da Piano di Progetto di seguito si riportano le quattro milestone prefissate prima delle quali si effettuerà una verifica del prodotto:

- Revisioni formali:
 - Revisione dei Requisiti (27/04/2015)
 - Revisione di Accettazione (06/07/2015)
- Revisioni di progresso:
 - Revisione di Progettazione (29/05/2015)
 - Revisione di Qualifica (18/07/2015)

Sarà necessario, infine, assicurarsi che ogni requisito sia tracciato consistentemente nel documento di Analisi dei Requisiti.

3.3 Responsabilità

I principali ruoli di responsabilità individuati sono:

- Amministratore di Progetto:
 - Assicura la funzionalità dell'ambiente di lavoro;
 - Redige i piani di gestione della qualità e ne verifica l'applicazione.
- Responsabile del progetto:
 - Assicura lo svolgimento delle attività di verifica;
 - Assicura il rispetto dei ruoli e delle competenze come descritti nel Piano di Progetto;
 - Approva e sancisce la distribuzione di un documento o di un file di codice;
 - Assicura il rispetto delle scadenze.

4 Risorse

4.1 Risorse necessarie:

4.1.1 Risorse umane

I ruoli necessari a garantire la qualità del prodotto sono:

- Responsabile di Progetto;
- Amministratore;
- Verificatore;
- Programmatore.

4.1.2 Risorse Hardware

Saranno necessari:

- Computer con installato software necessario allo sviluppo del progetto in tutte le sue fasi;
- Luoghi in cui svolgere riunioni, preferibilmente dotato di connessione ad Internet.

4.1.3 Risorse software

Saranno necessari:

- Strumenti per automatizzare i test;
- Framework per eseguire test di unità;
- Piattaforma di versionamento per la creazione e gestione di ticket;
- Debugger per i linguaggi di programmazione scelti;
- Browser come piattaforma di testing dell'applicazione da sviluppare;
- Strumenti per effettuare l'analisi statica del codice per misurare le metriche.

4.2 Risorse disponibili

Sono disponibili:

- Computer personali dei membri del gruppo;
- Computer presenti nelle aule informatiche del Dipartimento di Matematica;
- Aule disponibili per incontri nel Dipartimento di Matematica;
- Un dispositivo Raspberry Pi 2 Model B, utilizzato come server per programmi organizzativi e di testing.

4.2.1 Risorse software

Si rimanda alla sezione Strumenti 5.1.



5.1 Strumenti

- **Aspell**: correttore ortografico per documenti redatti in L^AT_EX;
- **Aptana**: scelto dal gruppo per la stesura del codice ha integrato al suo interno varie funzioni di debugging ed esecuzione del codice;
- **W3C validator**: sito che controlla la validità dei markup nei documenti scritti in HTML;
- **Jenkins**: sistema di per l'integrazione continua del codice e dei file latex;
- **jSHint**: tool di supporto per la ricerca di eventuali errori nel codice javascript;
- **Selenium IDE**: estensione di Firefox che permette di registrare test tramite browser;
- **ApacheBench**: strumento a linea di comando utilizzato per misurare l'efficienza di un server web ed in grado di simulare situazioni di sovraccarico della rete;
- **SpeedTracer**: plugin per Google Chrome che permette di verificare l'efficienza di un'applicazione web durante la sua esecuzione.

5.2 Tecniche

- **Analisi statica:** consiste nell'analizzare il codice tramite tools e letture senza tuttavia eseguirlo. Data la natura di questo tipo di analisi, è possibile applicarla anche per il controllo di tutti i documenti testuali prodotti. Si esegue applicando i due seguenti metodi:
 - **Inspection:** l'obiettivo di questa tecnica di analisi è l'individuazione di difetti attraverso la lettura mirata del codice. Un prerequisito per questa metodologia di verifica è la definizione di una lista di controllo che elenca le possibili sezioni o passaggi maggiormente soggetti ad errori. La verifica deve essere condotta da soggetti nettamente distinti dai programmatori. La correzione degli errori individuati va eseguita in ogni fase e documentata tramite un rapporto delle attività svolte;
 - **Walkthrough:** l'obiettivo di questa tecnica di analisi è l'individuazione di difetti eseguendo una lettura integrale di tutto il codice senza l'assunzione di presupposti. Viene eseguita da gruppi misti di ispettori e sviluppatori. Per evitare incomprensioni è importante che al termine della lettura gli elementi coinvolti discutano i difetti trovati e che nessuna persona possa coprire entrambi i ruoli allo stesso tempo. Al termine della fase di discussione si applicherà la fase di correzione dei difetti che apporterà le modifiche concordate. Anche in questo caso è importante tenere un rapporto delle attività svolte.

- **Analisi dinamica:** consiste nel verificare e validare il software o un suo componente osservandone il comportamento in esecuzione durante lo svolgimento di test. Tali test devono essere svolti in maniera ripetibile: significa che se eseguiti nello stesso ambiente e con gli stessi ingressi, devono produrre i medesimi risultati.
 - **Test di unità:** esamina la correttezza di piccole unità di codice, generalmente prodotte da un singolo programmatore, in modo da verificare che esse rispettino i loro requisiti. È possibile svolgerlo con un alto grado di parallelismo possibilmente servendosi di un automa;
 - **Test di integrazione:** verifica che l'integrazione delle unità che hanno superato il test precedente non produca problemi. Tali problemi, non potendo essere relativi alle singole unità, saranno da ricercare nell'interfaccia che le aggrega;
 - **Test di sistema:** accerta la copertura dei requisiti software individuati nell'analisi dei requisiti permettendo la validazione del sistema prodotto;
 - **Test di regressione:** stabilisce se modifiche all'implementazione di un programma alterano elementi precedentemente funzionanti. Per far ciò si eseguono nuovamente i test di unità e integrazione sulle parti modificate;
 - **Test funzionali:** mettono alla prova le funzionalità del sistema, simulando l'iterazione tra Utente e sistema;
 - **Test Prestazionali:** valutano le prestazioni dell'applicazione in molti modi e da molti punti di vista. Questo tipo di test mostra ciò che proverà l'utente in termini di caricamento e velocità del sito. Le prestazioni sono importanti anche per motivi SEO, in quanto un sito lento verrà analizzato molto meno frequentemente dai web crawler dei motori di ricerca;
 - **Test di collaudo:** attività formale supervisionata dal committente il cui buon esito comporta la possibilità di rilasciare il prodotto;

5.3 Metodi

Il gruppo ha deciso di utilizzare i seguenti metodi per applicare le tecniche sopra descritte, aiutandosi con gli strumenti elencati:

- **Documenti \LaTeX :**
 1. Rilettura approfondita;
 2. Controllo ortografico tramite lo strumento Aspell;
 3. Controllo dell'applicazione delle regole tipografiche espresse nel documento Norme di progetto;
 4. Verifica della corretta formattazione del file pdf prodotto.
- **Codice:** il codice verrà analizzato dagli strumenti integrati all'interno dell'IDE Aptana e dagli strumenti di sviluppo forniti dai singoli browser.
- **Schemi UML:**

1. data l'impossibilità di controllare la correttezza ortografica degli schemi, con Aspell è necessario esaminare attentamente e più volte i nomi, gli identificativi e i testi nei diagrammi;
2. controllo della correttezza degli identificativi dei casi d'uso rispetto alla nomenclatura stabilita nel documento *Norme di Progetto v.1.0.0* e rispetto alle sezioni dell'Analisi dei Requisiti in cui sono inseriti;
3. controllo della numerazione dei casi d'uso rispetto la loro gerarchia;
4. controllo che i casi d'uso soddisfino tutte le esigenze espresse nel capitolato.

5.4 Metriche

Il processo di verifica, per essere informativo, deve esse quantificabile. Le misure rilevate dal processo di verifica devono quindi essere basate su metriche stabilite a priori.

Una metrica è la misura di una qualche proprietà relativa ad una porzione di un documento-software allo scopo di fornire informazioni significative sulla qualità del codice prodotto.

Non bisogna tuttavia basarsi solamente sulle metriche, che sono solo indicatori a posteriori della bontà del lavoro svolto: un'importanza ancora maggiore la riveste il controllo sulla qualità del processo.

5.4.1 Metriche per il codice

- **Complessità Ciclomatica di McCabe:** è indicazione del numero di segmenti lineari in un metodo (ad esempio sezioni di codice senza ramificazioni), può quindi essere usato per determinare il numero di test necessari per ottenere una copertura completa dei possibili cammini.

Un metodo senza ramificazioni ha Complessità Ciclomatica di McCabe pari a 1; tale valore è incrementato ogni qualvolta si incontra una ramificazione.

Con “ramificazione” si intendono cicli, costrutti “if” e simili;

Secondo McCabe una complessità ciclomatica nel range 1-10 individua un codice semplice con pochi rischi, superato questo limite il codice diventa più complesso, instabile e difficilmente manutenibile;

- **Numero linee di codice:** rappresenta il numero di linee di codice all'interno di un blocco. Un indice elevato non rappresenta necessariamente un cattivo codice ma suggerisce la possibilità di estrarre metodi contenenti gruppi di istruzioni correlate, aumentando il livello di astrazione;

- **Indice di manutenibilità:** calcola un indice dal valore compreso tra 0 e 100 che rappresenta la facilità di manutenibilità del codice.

Un elevato valore indica un'ottima manutenibilità: un punteggio tra 20 e 100 indica che il codice ha manutenibilità buona; tra 10 e 19, una manutenibilità moderata; un punteggio tra 0 e 9 indica bassa manutenibilità;

- **Copertura del codice:** è indicazione di quanto codice sorgente sia stato testato. Un elevato indice di copertura indica che il codice sorgente è stato testato in profondità e che difficilmente può contenere dei bug.

Parametri utilizzati:

- Range-sufficiente: [60%-80%];
- Range-ottimale: [80%-95%].

5.4.2 Metriche per i documenti

- **Indice Gulpease:** misura l'indice di leggibilità di un testo tarato sulla lingua italiana. Rispetto ad altri ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. Permette di misurare la complessità dello stile di un documento. L'indice Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere. L'indice è calcolato secondo la seguente formula:

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

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.

Parametri utilizzati:

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



6.1 Comunicazione e risoluzione di anomalie

- Violazione delle norme tipografiche in un documento;
- Uscita dal range d'accettazione degli indici di misurazione;
- Incongruenza del prodotto con funzionalità presenti nell'analisi dei requisiti;
- Incongruenza del codice con il design del prodotto.

Quando viene rilasciata una nuova versione di un documento od un modulo, il Verificatore controlla il registro delle modifiche ed in base ad esso effettua una verifica alla ricerca di anomalie da correggere. Se ne trova, apre un ticket e lo comunica all'Amministratore; s'occuperà della correzione la persona che ha apportato la modifica al documento o modulo. Le nuove modifiche dovranno essere approvate dall'Amministratore.



7.1 Riassunto delle attività di verifica

È stato trovato anche qualche errore più grave, come il mancato rispetto delle regole di formattazione riportate nelle *Norme di Progetto v.1.0.0* e alcune mancanze all'interno del documento di *Analisi dei Requisiti v.1.0.0*.

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante la fase di **Analisi**. Un documento è considerato valido soltanto se rispetta le metriche descritte su 5.4.2.

Documento	Valore indice	Esito
<i>Piano di Progetto v.1.0.0</i>	?	?
<i>Analisi dei Requisiti v.1.0.0</i>	?	?
<i>Norme di Progetto v.1.0.0</i>	?	?
<i>Piano di Qualifica v.1.0.0</i>	?	?
<i>Studio di Fattibilità v.1.0.0</i>	?	?
<i>Glossario v.1.0.0</i>	?	?

Tab 3: Esiti verifica documenti, Analisi

Come si può notare dalla tabella, tutti gli indici Gulpease dei documenti rientrano nel range ottimale precedentemente definito e quindi i documenti redatti hanno raggiunto la leggibilità desiderata.



8 Verifica dei Requisiti

8.1 Verifica dei Requisiti Funzionali

Tab 4: Verifica dei Requisiti Funzionali

Test	Descrizione	Requisito
TS 1	Viene verificato che ci si possa registrare al sistema inserendo username e password	RF 1
TS 3	Viene verificato che ci si possa autenticare con username e password	RF 3
TS 4	Viene verificato che si possa creare una nuova presentazione vuota	RF 4
TS 7	Viene verificato che si possa passare in modalità modifica di una presentazione da desktop	RF 7
TS 7.1.1	Viene verificato che si possa inserire un nuovo frame nel piano della presentazione	RF 7.1.1
TS 7.1.1.1	Viene verificato che si possa scegliere il tipo di frame da inserire	RF 7.1.1.1
TS 7.1.4	Viene verificato che si possa spostare un frame in modalità modifica	RF 7.1.4
TS 7.1.7	Viene verificato che si possa passare in modalità modifica di un frame	RF 7.1.7
TS 7.1.7.1	Viene verificato che su possa inserire del testo all'interno di un frame	RF 7.1.7.1
TS 7.1.7.4	Viene verificato che si possa modificare del testo già presente all'interno di un frame	RF 7.1.7.4
TS 7.1.7.7	Viene verificato che si possa inserire un immagine all'interno del frame	RF 7.1.7.7



Test	Descrizione	Requisito
TS 7.1.7.10	Viene verificato che si possa modificare la dimensione di un immagine	RF 7.1.7.10
TS 7.1.7.13	Viene verificato che si possa inserire un video all'interno di un frame	RF 7.1.7.13
TS 7.1.7.16	Viene verificato che si possa modificare la dimensione di un video all'interno di un frame	RF 7.1.7.16
TS 7.1.7.19	Viene verificato che si possa spostare un elemento all'interno del frame	RF 7.1.7.19
TS 7.1.7.22	Viene verificato che si possa eliminare un elemento presente all'interno del frame	RF 7.1.7.22
TS 7.1.7.25	Verificare che si possa modificare il testo di un elemento scelta	RF 7.1.7.25
TS 7.1.7.28	Viene verificato che si possa modificare la dimensione di un frame	RF 7.1.7.28
TS 7.1.7.31	Viene verificato che si possa modificare la forma di un frame	RF 7.1.7.31
TS 7.1.7.34	Viene verificato che si possa modificare lo spessore del bordo di un frame	RF 7.1.7.34
TS 7.1.7.37	Viene verificato che si possa modificare il colore del bordo di un frame	RF 7.1.7.37
TS 7.1.7.40	Viene verificato che si possa modificare lo sfondo di un frame	RF 7.1.7.40, 7.1.7.43
TS 7.1.10	Viene verificato che si possa eliminare un frame dal piano di una presentazione	RF 7.1.10
TS 7.1.13	Viene verificato che si possa inserire un'immagine di sfondo in un'area della presentazione	RF 7.1.13
TS 7.1.16	Viene verificato che si possa inserire un colore di sfondo in un area della presentazione	RF 7.1.16

Test	Descrizione	Requisito
TS 7.1.19	Viene verificato che si possa definire un percorso di visualizzazione	RF 7.1.19
TS7.1.19.1	Viene verificato che si possa impostare un frame iniziale per il percorso di presentazione	RF7.1.19.1
TS7.1.19.4	Viene verificato che si possa definire una transizione tra due frame	RF7.1.19.4
TS7.1.19.7	Viene verificato che si possa definire una transizione scelta tra due frame	RF7.1.19.7
TS7.1.19.10	Viene verificato che si possa eliminare una transizione tra due frame	RF7.1.19.10
TS7.1.19.13	Viene verificato che si possa togliere un frame dal percorso di presentazione	RF7.1.19.13
TS7.1.22	Viene verificato che si possa assegnare un bookmark ad un frame	RF7.1.22
TS7.1.25	Viene verificato che si possa rimuovere un bookmark da un frame	RF7.1.25
TS7.1.28	Viene verificato che si possa modificare la velocità di transizione tra due frame consecutivi	RF7.1.28
TS7.1.31	Viene verificato che si possa impostare un effetto di transizione tra due frame consecutivi	RF7.1.31
TS7.1.34	Viene verificato che si possa impostare il tempo di attesa tra due frame consecutivi durante la riproduzione automatica	RF7.1.34
TS7.1.37	Viene verificato che si possa annullare e ripristinare una modifica appena effettuata	RF7.1.37, 7.1.40

Test	Descrizione	Requisito
TS 10	Viene verificato passare in modalità modifica di una presentazione da mobile	RF 10
TS 10.1	Viene verificato che si possa editare testo da mobile all'interno di un frame	RF 10.1
TS 10.4	Viene verificato che si possa modificare da mobile il testo presente all'interno di un frame	RF 10.4
TS 10.3	Viene verificato che si possa annullare da mobile una modifica appena effettuata	RF 10.3
TS 10.5	Viene verificato che si possa assegnare un bookmark ad un frame da mobile	RF 10.5
TS 10.8	Viene verificato che si possa rimuovere un bookmark ad un frame da mobile	RF 10.8
TS 13	Viene verificato che si possa caricare un'immagine dal proprio File System alla propria parte dedicata alle immagini	RF 13
TS 16	Viene verificato che si possano eliminare dal server le immagini caricate	RF 16
TS 19	Viene verificato che si possa creare nuove cartelle e spostare i file all'interno delle cartelle	RF 19
TS 25	Viene verificato che si possano organizzare le proprie presentazioni con una struttura a cartelle	RF 25
TS 31	Viene verificato che si possano spostare le proprie infografiche all'interno della cartella dedicata sul server	RF 31
TS 34	Viene verificato che si possa eliminare dal server una presentazione creata	RF 34

Test	Descrizione	Requisito
TS 37	Viene verificato che si possa eliminare dal server un'infografica creata	RF 37
TS 43	Viene verificato che si possa modificare la propria password di accesso al sistema	RF 43
TS 46	Viene verificato che si possa scaricare in locale un'infografica creata sul server	RF 46
TS 49	Viene verificato che si possa salvare in locale una presentazione creata sul server	RF 49
TS 52	Viene verificato che si possa rimuovere una presentazione salvata in locale	RF 52
TS 55	Viene verificato che si possa eseguire una presentazione salvata sul server	RF 55
TS 58	L'utente deve essere in grado di eseguire una presentazione salvata in locale	RF 58
TS 61.1	Viene verificato che si possa eseguire una presentazione in modalità manuale	RF 61.1
TS 61.1.1	Viene verificato che durante la presentazione si possa passare al frame successivo o al precedente	RF 61.1.1
TS 61.1.4	Viene verificato che si possa selezionare un elemento scelta se presente nel frame	RF 61.1.4
TS 61.1.7	Viene verificato che si possa passare al frame con bookmark successivo o precedente	RF 61.1.7
TS 61.1.10	Viene verificato che si possa passare da un frame visualizzato al suo frame contenitore	RF 61.1.10
TS 61.1.13	Viene verificato che si possa eseguire dello zoom in una parte qualsiasi del frame	RF 61.1.13

Test	Descrizione	Requisito
TS 61.1.16.1	Viene verificato che si possa far partire l'esecuzione di un video all'interno di un frame	RF 61.1.16.1
TS 61.1.16.4	Viene verificato che si possa sospendere e poi riprendere l'esecuzione di un video all'interno di un frame	RF 61.1.16.4
TS 61.1.16.7	Viene verificato che si possa eseguire un video da un punto qualsiasi dello stesso	RF 61.1.16.7
TS 61.1.16.10	Viene verificato che si possa interrompere l'esecuzione di un video	RF 61.1.16.10
TS 61.4	Viene verificato che si possa eseguire una presentazione in modalità automatica	RF 61.4
TS 61.4.1	Viene verificato che si possa chiudere una presentazione in esecuzione automatica	RF 61.4.1
TS 61.4.4	Viene verificato che si possa sospendere e riavviare una presentazione in esecuzione automatica	RF 61.4.4
TS 61.4.7	Viene verificato che si possa impostare la velocità di riproduzione della presentazione	RF 61.4.7
TS 61.4.10	Viene verificato che si possa saltare la riproduzione di un video nel frame visualizzato	RF 61.4.10
TS 61.7	Viene verificato che si possa passare da presentazione automatica a presentazione manuale e viceversa	RF 61.7, 61.10
TS 64	Viene verificato che si possa effettuare il logout dal server	RF 64
TS 67.1	Viene verificato che l'amministratore possa inserire dei template di presentazioni	RF 67.1

Test	Descrizione	Requisito
TS 67.4	Viene verificato che l'amministratore possa inserire template di infografiche	RF 67.4
TS 67.7	Viene verificato che l'amministratore possa inserire elementi grafici	RF 67.7
TS 67.10	Viene verificato che l'amministratore possa eliminare un template	RF 67.10
TS 67.13	Viene verificato che l'amministratore possa annullare l'ultima eliminazione di un template	RF 67.13
TS 70.1	Viene verificato che si possa selezionare una presentazione da cui produrre l'infografica	RF 70.1
TS 70.4	Viene verificato che si possa selezionare template di infografica	RF 70.4
TS 70.7	Viene verificato che si possano selezionare gli elementi del	RF 70.7
TS 70.10	Viene verificato che si possa passare in modalità modifica di un'infografica	RF 70.10
TS 70.10.1	Viene verificato che si possa modificare un elemento di un infografica	RF 70.10.1
TS 70.10.1.1	Viene verificato che si possano modificare le dimensioni di un immagine	RF 70.10.1.1
TS 70.10.1.4	Viene verificato che si possa modificare un elemento testo	RF 70.10.1.4
TS 70.10.1.4.1	Viene verificato che si possa modificare il font del testo	RF 70.10.1.4.1
TS 70.10.1.4.4	Viene verificato che si possa modificare la dimensione del carattere	RF 70.10.1.4.4
TS 70.10.1.4.7	Viene verificato che si possa modificare lo stile del testo	RF 70.10.1.4.7

Test	Descrizione	Requisito
TS 70.10.1.4.10	Viene verificato che si possa modificare il colore della scritta del testo	RF 70.10.1.4.10
TS 70.10.1.4.13	Viene verificato che si possa modificare il colore dell'evidenziazione del testo	RF 70.10.1.4.13
TS 70.10.1.7	Viene verificato che si possa cambiare la posizione dell'elemento	RF 70.10.1.7
TS 70.10.4	Viene verificato che si possa rimuovere lo sfondo dell'infografica	RF 70.4.4
TS 70.10.7	Viene verificato che si possa inserire uno sfondo nell'infografica	RF 70.10.7
TS 70.10.10	Viene verificato che si possa inserire un elemento immagine nell'infografica	RF 70.10.10
TS 70.10.13	Viene verificato che si possa inserire del testo nell'infografica	RF 70.10.13
TS 70.10.16	Viene verificato che si possa inserire un frame nella sua interezza presente nella presentazione	RF 70.10.16
TS 70.10.19	Viene verificato che si possano eliminare elementi immagini o testuali	RF 70.10.19
TS 70.13	Viene verificato che si possa salvare l'infografica nel suo spazio	RF 70.13
TS 70.16	Viene verificato che si possa annullare e ripristinare una modifica appena effettuata	RF 70.16
TS 70.19	Viene verificato che si possa esportare un'infografica in formato stampabile	RF 70.19
TS 73	Viene verificato che si possa creare un'infografica	RF 73

8.2 Verifica dei Requisiti di Qualità e Vincoli

Tab 5: Verifica dei Requisiti di Qualità e Vincoli

Descrizione	Requisito
Viene verificato che ogni funzionalità dell'applicazione sia documentato	RQ 1, 7
Viene verificato che sia disponibile un tutorial interattivo per la creazione delle presentazioni	RQ4
Viene verificato che sia disponibile della documentazione sui test eseguiti	RQ10
Viene verificato che il sistema dovrà offrire la possibilità di eseguire offline le presentazioni	RQ13
Viene verificato che il sistema sia funzionante su dispositivi Desktop e Mobile (Android, Ios, Windows Phone)	RQ16
Viene verificato che il sistema segua le linea guida del material design fornite dalla Google	RV 1