

19-08-2015



Piano di Qualifica

Informazioni sul documento

Nome Documento	Piano di Qualifica
Versione	3.0.0
Stato	<i>Formale</i>
Uso	<i>Esterno</i>
Data Creazione	20-03-2015
Data Ultima Modifica	19-08-2015
Redazione	Venturelli Giovanni
Approvazione	Fossa Manuel
Verifica	Petrucci Mauro
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
3.0.0	Fossa Manuel	19-08-2015	Approvazione del documento
2.8.0	Petrucci Mauro	18-08-2015	Aggiunte metriche per il codice §A.4
2.7.0	Petrucci Mauro	18-08-2015	Aggiornamento test §B.3
2.6.0	Venturelli Giovanni	27-06-2015	Aggiornamento capitolo Obiettivi di Qualità
2.5.0	Gabelli Pietro	16-06-2015	Apportate correzioni automatiche
2.4.0	Busetto Matteo	11-06-2015	Aggiunta test d'Integrazione
2.3.0	Venturelli Giovanni	11-06-2015	Incremento capitolo Obiettivi di Qualità
2.2.0	Gabelli Pietro	11-06-2015	Apportate correzioni automatiche
2.1.0	Gabelli Pietro	11-06-2015	Inseriti test d'Integrazione
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	Tollet 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

Storico

pre-RR

Versione 1.0.0	Nominativo
Redazione	Fossa Manuel, Petrucci Mauro
Verifica	Gabelli Pietro
Approvazione	Tollot Pietro

Tab 2: Storico ruoli pre-RR

$$\mathbf{RR} \rightarrow \mathbf{RP}$$

Versione 2.0.0	Nominativo
Redazione	Venturelli Giovanni
Verifica	Gabelli Pietro
Approvazione	Petrucci Mauro

Tab 3: Storico ruoli RR \rightarrow RP

$$\mathbf{RP} \rightarrow \mathbf{RQ}$$

Versione 3.0.0	Nominativo
Redazione	Venturelli Giovanni
Verifica	Petrucci Mauro
Approvazione	Fossa Manuel

Tab 4: Storico ruoli RP \rightarrow RQ

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	10
2.2.1	Funzionalità	10
2.2.2	Affidabilità	11
2.2.3	Efficienza	11
2.2.4	Usabilità	11
2.2.5	Manutenibilità	11
2.2.6	Portabilità	12
2.3	Procedure di controllo di qualità di processo	12
3	Visione generale delle strategie di verifica	14
3.1	Organizzazione	14
3.2	Pianificazione strategica e temporale	14
3.3	Responsabilità	15
3.4	Risorse	15
3.5	Tecniche di analisi	16
3.6	Metriche	17
3.6.1	metriche _g per la progettazione	17
3.6.2	metriche _g per il codice	17
3.6.3	metriche _g per i documenti	19
Appendice A Riassunto delle attività di verifica		20
A.1	Revisione dei Requisiti	20
A.2	Documenti	20
A.3	Progettazione Architetutturale	20
A.4	Progettazione di dettaglio e codifica	21
Appendice B Pianificazione dei test		23
B.1	Test di sistema	23
B.2	Test d'integrazione	33
B.3	Test d'unità	35



1	Rappresentazione del modello ISO/IEC 9126:2001	10
2	Schema PDCA	12
3	Sequenza d'integrazione delle componenti	33

1	Versionamento del documento	1
2	Storico ruoli pre-RR	2
3	Storico ruoli RR -> RP	2
4	Storico ruoli RP -> RQ	2
5	Esiti verifica documenti, Analisi	20
6	Tabella accoppiamento afferente ed efferente delle componenti	21
7	Descrizione dei test di sistema per i Requisiti _g Funzionali	23
8	Descrizione dei test di sistema per i Requisiti _g di Qualità e Vincoli	32
9	Descrizione dei test di Integrazione	34
10	Descrizione dei test di Unità per il Server _g	35
11	Descrizione dei test di Unità per ServerRelation; classe: fileServerRelation	39
12	Descrizione dei test di Unità per ServerRelation; classe: serverRelation	40
13	Descrizione dei test di Unità per ServerRelation; classe: Loader	42
14	Descrizione dei test di Unità	42
15	Descrizione dei test di Unità per SlideShowElements	43
16	Descrizione dei test di Unità per InsertEditRemove	43

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.3.0.0.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.3.0.0.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

- Piano di Progetto_g: [PianoDiProgetto_v.3.0.0.pdf](#);
- Slide dell'insegnamento Ingegneria del Software_g modulo A:
<http://www.math.unipd.it/~tullio/IS-1/2014/> ;
- SWEBOK – Version 3 (2004): capitolo 11 – Software_g Quality
<http://www.computer.org/portal/web/swebok/html/ch11>;
- Ingegneria del Software_g - Ian Sommerville - 9a Edizione (2010):
 - Capitolo 24 - Gestione della qualità;
 - Capitolo 26 - Miglioramento dei Processi_g.
- Standard ISO_g /IEC TR 15504: Software_g process assessment
http://en.wikipedia.org/wiki/ISO/IEC_15504;

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



2.1 Qualità di processo

Tutti i Processi_g dovranno quindi essere sottoposti a valutazione in modo da verificarne la qualità ed eventualmente facilitarne il miglioramento. A tale scopo lo SPICE_g definisce nove attributi di Processo_g per effettuare una migliore valutazione:

- I lavori identificati come input al Processo_g (input work products);
- I lavori identificati come output del Processo_g (output work products);
- Le azioni compiute per trasformare gli input work products in output work products.

L'attuazione di un Processo_g è pianificata e controllata al fine di generare risultati che siano adeguatamente documentati, controllati e verificati;

L'attuazione di un Processo_g si basa su approcci standardizzati;

Il Processo_g può contare su adeguate risorse umane_g, di infrastrutture, ecc. per essere attuato;

I risultati conseguiti e le misure rilevate durante l'attuazione di un Processo_g sono utilizzati per assicurarsi che l'attuazione di tale Processo_g supporti efficacemente il raggiungimento di obiettivi specifici;

Un Processo_g è controllato tramite la raccolta, analisi ed utilizzo delle misure di prodotto e di Processo_g rilevate, con l'obbiettivo di correggere, se necessario, le sue modalità di attuazione;

Le modifiche alla definizione, gestione e attuazione di un Processo_g sono controllate;

- **Livello 5** - Processo_g ottimizzante: il Processo_g è predicibile ed in grado di adattarsi per raggiungere obiettivi specifici e rilevanti per l'organizzazione. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di "Process change" e "Continuous integration".

L'applicazione dello standard ISO/IEC 15504 porta a benefici sia agli sviluppatori del Software_g che ai suoi utilizzatori o acquirenti. Per gli sviluppatori porta vantaggi nell'ottimizzazione dell'uso delle Risorse_g, un contenimento dei costi, una maggiore tempestività di consegna del prodotto ultimato, migliore stima dei rischi e degli impegni e la possibilità di confrontarsi con delle best practice. Per gli utenti invece abbiamo una maggior facilità nella selezione dei fornitori, una migliore valutazione dei rischi di Progetto_g, controllo dello stato di avanzamento in corso d'opera, riduzione dei costi di correzione degli errori ed un controllo dei rischi e delle varianti in corso d'opera.

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 25010:2011. Questo documento fornisce un modello per valutare la qualità esterna (nell'ambiente di utilizzo) ed interna (indipendente dall'ambiente) di un Software_g, individuando sei caratteristiche principali atte a rendere il prodotto qualitativamente accettabile.

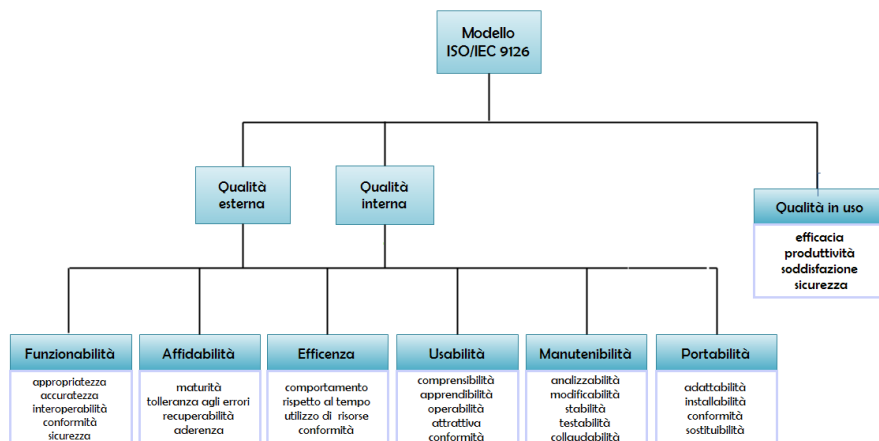


Fig 1: Rappresentazione del modello ISO/IEC 9126:2001

2.2.1 Funzionalità

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

Quantificazione del raggiungimento dell'obiettivo di qualità: la misurazione del raggiungimento di questo obiettivo si calcolerà verificando la quantità di Requisiti_g soddisfatti che

avranno un riscontro in elementi_g funzionanti nell'applicazione prodotta. La soglia di sufficienza sarà quindi data dal soddisfacimento di tutti i Requisiti_g obbligatori previsti dal capitolato d'appalto.

2.2.2 Affidabilità

È un Requisito_g non funzionale che indica la capacità del Software_g 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.

Quantificazione del raggiungimento dell'obiettivo di qualità: la misurazione del raggiungimento di questo obiettivo si calcolerà confrontando il numero di esecuzioni totale con quelle andate a buon fine e che hanno mantenuto un livello di prestazioni tali da poter permettere l'utilizzo previsto del prodotto

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, richieste.

Quantificazione del raggiungimento dell'obiettivo di qualità: 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.

Quantificazione del raggiungimento dell'obiettivo di qualità: Questa caratteristica non è facilmente misurabile in quanto non esistono metriche_g per quantificarla, perciò si farà affidamento alle linee guida del material design_g 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.

Quantificazione del raggiungimento dell'obiettivo di qualità: Per la misurazione di questo Requisito_g si fa riferimento alle metriche_g descritte nella sezione 3.6.

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.

Quantificazione del raggiungimento dell'obiettivo di qualità: 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, che permette di riconoscere lo stato di avanzamento di un Progetto_g fornendo un metodo di lavoro logico e sistematico.

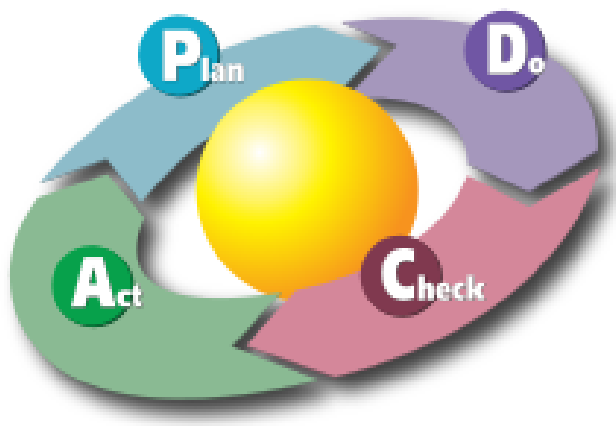


Fig 2: Schema PDCA

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

- **Plan:** si stabiliscono obiettivi e Processi_g necessari ad ottenere risultati conformi agli obiettivi attesi;
- **Do:** si implementa il piano, si esegue il Processo_g 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.

Una descrizione di come il gruppo applicherà il PDCA è riportata nelle *Norme di Progetto v.3.0.0*(§5.1).

3 Visione generale delle strategie di verifica

3.1 Organizzazione

Ogniquale volta avvenga un cambiamento sostanziale nello sviluppo del prodotto, si istanzia il Processo_g di verifica.

Nello specifico durante ogni fase (Analisi, Progettazione, Realizzazione e Validazione_g) 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_g di Codice_g ;
- 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 [NormeDiProgetto_v.3.0.0.pdf](#);
- Conclusione della modifica sostanziale di un File_g di Codice_g , 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 [NormeDiProgetto_v.3.0.0.pdf](#).

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_g 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_g .

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

DIFETTO \rightarrow *ERRORE* \rightarrow *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_g il ramo di Codice_g che le contiene.

È 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

Al fine di rendere sistematica l'attività di verifica, per poter rispettare le scadenze fissate nel Piano di Progetto_g ed evitare la propagazione di errori all'interno dei documenti o di File_g di Codice_g prima della loro verifica, la loro redazione sarà anticipata da una fase di studio preliminare.

Questa fase permetterà di ridurre la necessità di grossi interventi nelle fasi successive, quando



- Revisioni formali:
 - Revisione dei Requisiti_g (28/04/2015)
 - Revisione di Accettazione (06/07/2015)
- Revisioni di progresso:
 - Revisione di Progettazione (29/05/2015)
 - Revisione di Qualifica (18/06/2015)

3.3 Responsabilità

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

Per assicurare che gli obiettivi qualitativi vengano raggiunti è necessario l'utilizzo di Risorse_g sia umane che tecnologiche. Per una dettagliata descrizione dei ruoli e delle loro responsabilità fare riferimento alle Norme di Progetto_g. Per Risorse_g tecniche e tecnologiche sono da intendersi tutti gli strumenti Software_g e hardware che il gruppo intende utilizzare per attuare le attività di verifica su Processi_g e prodotti. Affinché il lavoro dei Verificatori venga agevolato si sono predisposti numerosi strumenti automatici che eseguono controlli sistematici sui prodotti generati. Tali strumenti sono descritti in modo accurato nelle Norme di Progetto_g.



3.5 Tecniche di analisi

- **Analisi statica:** consiste nell'analizzare il Codice_g 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:
 - **Walkthrough:** Si svolge effettuando una lettura critica a largo spettro. È una tecnica che viene utilizzata soprattutto nelle prime attività del Progetto_g, quando ancora non è presente una adeguata esperienza da parte dei membri del gruppo, che permetta di attuare una verifica più mirata e precisa. Con l'utilizzo di questa tecnica, il Verificatore sarà in grado di stilare una lista di controllo con gli errori più frequenti in modo da favorire il miglioramento di tale attività nelle fasi future. Questa è un'attività onerosa e collaborativa che richiede l'intervento di più persone per essere efficace ed efficiente. Dopo una prima fase di lettura e individuazione degli errori, segue una fase di discussione con la finalità di esaminare i difetti riscontrati e di proporre le dovute correzioni. L'ultima fase consiste nel correggere gli errori rilevati e nello scrivere un rapporto che elenchi le modifiche effettuate.
 - **Inspection:** Questa tecnica consiste nell'analisi mirata di alcune parti del documento o del Codice_g che sono ritenute fonti maggiori di errore. La *lista di controllo*, che deve essere seguita per svolgere efficacemente questo Processo_g, deve essere redatta anticipatamente ed è frutto dell'esperienza maturata dai verificatori attraverso la tecnica di Walkthrough. L'Inspection è una strategia più rapida del Walkthrough in quanto consente l'analisi di alcune parti dei prodotti ritenute critiche dalla checklist e non necessità della lettura integrale dei documenti in oggetto. Diversamente dal Walkthrough, tale tecnica viene svolta esclusivamente dai verificatori che dopo aver individuato gli errori procedono alla loro correzione e alla redazione di un rapporto di verifica che tenga traccia del lavoro svolto. Durante l'applicazione del Walkthrough ai documenti, sono state riportate le tipologie di errori più frequenti. La *lista di controllo* risultante è in appendice delle *Norme di Progetto v.3.0.0*.
- **Analisi dinamica:** consiste nel verificare e validare il Software_g 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_g, generalmente prodotte da un singolo programmatore, in modo da verificare che rispettino i Requisiti_g. Può essere svolto con un alto grado di parallelismo 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_g Software_g individuati nell'analisi dei Requisiti_g permettendo la Validazione_g del sistema prodotto;
 - **Test di regressione:** stabilisce se modifiche all'implementazione di un Programma_g alterano elementi_g 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_g tra utente e sistema;
- **Test di collaudo:** attività formale supervisionata dal Committente_g il cui buon esito comporta la possibilità di rilasciare il prodotto;

3.6 Metriche

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

Una Metrica_g è la misura di una proprietà relativa ad una porzione di un documento Software_g, allo scopo di fornire informazioni significative sulla qualità del Codice_g prodotto.

Per valutare la bontà del lavoro svolto non è sufficiente basarsi solo sulle metriche_g, che sono solamente degli indicatori valutati a posteriori, perché un'importanza ancora maggiore la riveste il controllo sulla qualità del Processo_g.

3.6.1 metriche_g per la progettazione

- **Numero di classi, coesione tra di esse e peso:** il peso di una classe è identificato dalla somma della complessità ciclomatica di tutti i metodi appartenenti alla classe;
- **Complessità di flusso:** misura la quantità di informazioni in entrata ed uscita da una Funzione_g (fan-in e fan-out).
 - **Fan-in:** numero di moduli che passano informazioni dentro al modulo in esame;
 - **Fan-out:** numero di moduli a cui il modulo in esame passa informazioni.

Il valore è calcolato come:

$$(lunghezza\ funzione)^2 \times fan - in \times fan - out.$$

Un Fan-in elevato indica un buon design strutturale: significa che il modulo in esame viene usato frequentemente, con conseguente riuso del modulo e riduzione della ridondanza del Codice_g.

Un Fan-out elevato indica un elevato accoppiamento tra moduli: significa che un modulo ha molte dipendenze da altri moduli e ciò mostra un povero design strutturale; implica anche un aumento dei costi di manutenibilità: ogni cambiamento di un modulo scatena la manutenzione dei moduli dipendenti.

3.6.2 metriche_g per il codice

- **Complessità Ciclomatica di McCabe:** è indicazione del numero di segmenti lineari in un metodo (ad esempio sezioni di Codice_g 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 ogniqualevolta 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_g semplice con pochi rischi, superato questo limite il Codice_g diventa più complesso, instabile e difficilmente manutenibile;

- **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_g di Halstead_g non è solamente un indice di complessità, ma identifica le proprietà misurabili del Software_g e le relative relazioni. Si basa sull'osservazione che una Metrica_g 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_g 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_g. 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_g esegue troppe operazioni.

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

Parametri utilizzati

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

- **Indice di manutenibilità:** Questa Metrica_g 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_g 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 fasi di manutenzione e incremento del Codice_g:

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

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

Parametri utilizzati:

- Range-sufficiente: [60%-80%];
- Range-ottimale: [80%-100%].

3.6.3 metriche_g per i documenti

- **Indice Gulpease:** misura l'indice di leggibilità di un testo; è tarato sulla lingua italiana. Rispetto ad altri indici 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].



A.1 Revisione dei Requisiti

È stato trovato anche qualche errore più grave, come il mancato rispetto delle regole di formattazione riportate nelle *Norme di Progetto v.3.0.0* e alcune mancanze all'interno del documento di *Analisi dei Requisiti v.3.0.0*.

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_g descritte su §3.6.3.

Tab 5: Esiti verifica documenti, Analisi

A.3 Progettazione Architeturale

Viene qui riportata una tabella riassuntiva che riporta il calcolo dei parametri di accoppiamento afferente ed efferente per i componenti individuati nella progettazione.

Componente	Afferente	Efferente
Premi::Model	4	5
Premi::Model::SlideShow	1	3
Premi::Model::SlideShow::SlideShowActions::InsertEditRemove	21	6
Premi::Model::SlideShow::SlideShowElements	3	7
Premi::Model::SlideShow::SlideShowActions::Command	2	21
Premi::Model::ServerRelations	12	3
Premi::Model::ServerRelations::Loader	3	1
Premi::Model::ServerRelations::AccessControl	4	1
Premi::Model::ServerRelations::DBConsistency	0	2
Premi::Model::ServerRelations::Presenter	5	4

Tab 6: Tabella accoppiamento afferente ed efferente delle componenti

Come si può vedere dalla tabella, l'accoppiamento efferente è generalmente molto basso e quindi positivo, ad eccezione del package Command, per il quale però questo è accettato a causa della natura intrinseca di tale componente. L'accoppiamento afferente mostra invece la stabilità richiesta dalle classi del Model.

A.4 Progettazione di dettaglio e codifica

Vengono qui riportate le misure rilevate con le metriche_g sull'analisi statica_g e dinamica del Codice_g. Per ogni Metrica_g si riportano i valore calcolati mantenendo una separazione tra back end e front-end. Per una descrizione delle metriche_g si rimanda alla sezione §3.6.2

Complessità ciclomatica

- Back-end

- Medio: 1.2
- Massimo: 3

- Front-end

- Medio: 2.4
- Massimo: 12

I valori del Back-end_g rispettano i parametri ottimali prefissati al contrario il Front-end ha una complessità ciclomatica troppo elevata.

Halstead_g Volume

- Back-end

- Medio: 53.3
- Massimo: 228

- Front-end

- Medio: 326.6
- Massimo: 1304

I Valori sia del Back-end_g che del front-end sono accettabili e rispettano i parametri prefissati.

Indice di manutenibilità

- Back-end

- Medio: 102.3
- Massimo: 153.69

- Front-end

- Medio: 95.6
- Massimo: 150.10

I valori sia del Back-end_g che del front-end sono accettabili e rispettano i parametri prefissati.

Copertura del codice

- **Back-end:**93.29 %
- **Front-end:**60.13 %

Il front-end rientra strettamente nella sufficienza della copertura del Codice_g questo a causa della difficoltà nel testare i controller ed ottenere una copertura elevata.

Il Back-end_g rientra nei parametri ottimali prefissati.

B Pianificazione dei test

Si descrivono di seguito tutti i test di Validazione_g, sistema ed integrazione previsti, prevedendo un aggiornamento futuro per i test di unità. Per le tempistiche di esecuzione dei test si faccia riferimento al [PianoDiProgetto_v.3.0.0.pdf](#).

B.1 Test di sistema

In questa sezione vengono descritti i test di sistema che permettono di verificare il comportamento dinamico del sistema completo rispetto ai Requisiti_g descritti nell' [AnalisiDeiRequisiti_v.3.0.0.pdf](#). I test di sistema riportati sono quelli relativi ai Requisiti_g Software_g individuati e pertanto meritevoli di un test.

Tab 7: Descrizione dei test di sistema per i Requisiti Funzionali

Test	Descrizione	Requisito
TS 1	Viene verificato che ci si possa registrare al sistema inserendo username e password	RF 1
TS 1.1	Viene verificato che si possa immettere uno username univoco	RF 1.1
TS 1.2	Viene verificato che si possa immettere una password valida	RF 1.2
TS 3	Viene verificato che ci si possa autenticare con username e password	RF 3
TS 3.1	Viene verificato che si possa immettere uno username	RF 3.1
TS 3.2	Viene verificato che si possa immettere una password	RF 3.2
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 _g	RF 7
TS 7.1	Viene verificato che si possa inserire un nuovo Frame _g nel piano della presentazione _g	RF 7.1

Test	Descrizione	Requisito
TS 7.1.1	Viene verificato che si possa scegliere il tipo di Frame _g da inserire	RF 7.1.1
TS 7.4	Viene verificato che si possa spostare un Frame _g in modalità modifica	RF 7.4
TS 7.7	Viene verificato che si possa passare in modalità modifica di un Frame _g	RF 7.7
TS 7.7.1	Viene verificato che si possa inserire del testo all'interno di un Frame _g	RF 7.7.1
TS 7.7.4	Viene verificato che si possa modificare del testo già presente all'interno di un Frame _g	RF 7.7.4
TS 7.7.7	Viene verificato che si possa inserire un'immagine all'interno del Frame _g	RF 7.7.7
TS 7.7.10	Viene verificato che si possa modificare la dimensione di un'immagine	RF 7.7.10
TS 7.7.13	Viene verificato che si possa inserire un video o un audio all'interno di un Frame _g	RF 7.7.13
TS 7.7.16	Viene verificato che si possa modificare la dimensione di un video all'interno di un Frame _g	RF 7.7.16
TS 7.7.19	Viene verificato che si possa spostare un Elemento _g all'interno del Frame _g	RF 7.7.19
TS 7.7.25	Verificare che si possa inserire un Elemento scelta _g in un Frame _g	RF 7.7.25
TS 7.7.28	Viene verificato che si possa modificare un Elemento scelta _g	RF 7.7.28
TS 7.7.31	Viene verificato che si possa modificare la forma di un Frame _g	RF 7.7.31

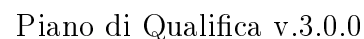
Test	Descrizione	Requisito
TS 7.7.34	Viene verificato che si possa modificare la dimensione di un Frame _g	RF 7.7.34
TS 7.7.37	Viene verificato che si possa modificare lo spessore del bordo di un Frame _g	RF 7.7.37
TS 7.7.40	Viene verificato che si possa modificare il colore del bordo di un Frame _g	RF 7.7.40
TS 7.7.43	Viene verificato che si possa modificare lo sfondo di un Frame _g	RF 7.7.43
TS 7.7.46	Viene verificato che si possa ruotare un Frame _g	RF 7.7.46
TS 7.10	Viene verificato che si possa eliminare un Frame _g dal piano di una presentazione	RF 7.10
TS 7.13	Viene verificato che si possa inserire un'immagine di sfondo in un'area della presentazione	RF 7.13
TS 7.16	Viene verificato che si possa inserire un colore di sfondo in un'area della presentazione	RF 7.16
TS 7.19	Viene verificato che si possa definire un Percorso _g di visualizzazione	RF 7.19
TS 7.19.1	Viene verificato che si possa impostare un Frame _g iniziale per il Percorso _g di presentazione	RF7.19.1
TS 7.19.4	Viene verificato che si possa definire una transizione tra due Frame _g	RF 7.19.4
TS 7.19.10	Viene verificato che si possa eliminare una transizione tra due Frame _g	RF 7.19.10
TS 7.19.13	Viene verificato che si possa togliere un Frame _g dal Percorso _g di presentazione	RF 7.19.13

Test	Descrizione	Requisito
TS 7.22	Viene verificato che si possa assegnare un Bookmark _g ad un Frame _g	RF 7.22
TS 7.25	Viene verificato che si possa rimuovere un Bookmark _g da un Frame _g	RF 7.25
TS 7.28	Viene verificato che si possa modificare la velocità di transizione tra due Frame _g consecutivi	RF 7.28
TS 7.31	Viene verificato che si possa impostare un effetto di transizione tra due Frame _g consecutivi	RF 7.31
TS 7.34	Viene verificato che si possa impostare il tempo di attesa tra due Frame _g consecutivi durante la riproduzione automatica	RF 7.34
TS 7.37	Viene verificato che si possa inserire un Elemento _g SVG in un Frame _g o nel piano della presentazione _g	RF 7.37
TS 7.40.1	Viene verificato che si possa modificare le dimensioni di un Elemento _g SVG	RF 7.40.1
TS 7.40.2	Viene verificato che si possa modificare il colore di un Elemento _g SVG	RF 7.40.2
TS 7.43	Viene verificato che si possa eliminare un Elemento _g	RF 7.43
TS 7.46	Viene verificato che si possa ruotare un Elemento _g	RF 7.46
TS 10	Viene verificato il poter passare in modalità modifica di una presentazione da mobile	RF 10
TS 10.1	Viene verificato che si possa editare da mobile il testo all'interno di un Frame _g	RF 10.1

Test	Descrizione	Requisito
TS 10.4	Viene verificato che si possa modificare da mobile il testo presente all'interno di un Frame _g	RF 10.4
TS 10.5	Viene verificato che si possa assegnare un Bookmark _g ad un Frame _g da mobile	RF 10.5
TS 10.8	Viene verificato che si possa rimuovere un Bookmark _g ad un Frame _g da mobile	RF 10.8
TS 13	Viene verificato che si possa caricare un File _g media presente in locale nel Server _g	RF 13
TS 16	Viene verificato che si possano eliminare dal Server _g i File _g media caricati	RF 16
TS 17	Viene verificato che si possano rinominare i File _g media presenti sul Server _g	RF 17
TS 19	Viene verificato che si possano rinominare le presentazioni salvate nel Server _g	RF 19
TS 25	Viene verificato che si possano rinominare le Infografiche _g salvate nel Server _g	RF 25
TS 31	Viene verificato che si possa eliminare dal Server _g un'Infografica _g creata	RF 31
TS 34	Viene verificato che si possa eliminare dal Server _g una presentazione creata	RF 34
TS 35	Viene verificato che si possano visualizzare le presentazioni salvate sul Server _g	RF 35
TS 36	Viene verificato che si possano visualizzare le Infografiche _g salvate sul Server _g	RF 36

Test	Descrizione	Requisito
TS 37	Viene verificato che si possano visualizzare i File _g media salvati sul Server _g	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 _g creata sul Server _g	RF 46
TS 49	Viene verificato che si possa salvare in locale una presentazione creata sul Server _g	RF 49
TS 52	Viene verificato che si possa rimuovere una presentazione salvata in locale	RF 52
TS 55	Viene verificato che si possa annullare una modifica non voluta	RF 55
TS 58	Viene verificato che si possa ripristinare una modifica annullata precedentemente	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 _g successivo o al precedente	RF 61.1.1
TS 61.1.4	Viene verificato che si possa selezionare un Elemento scelta _g se presente nel Frame _g	RF 61.1.4
TS 61.1.7	Viene verificato che si possa passare al Frame _g con Bookmark _g successivo o precedente	RF 61.1.7
TS 61.1.10	Viene verificato che si possa passare da un Frame _g visualizzato al suo Frame _g contenitore	RF 61.1.10
TS 61.1.13	Viene verificato che si possa zoomare in una parte qualsiasi del Frame _g	RF 61.1.13

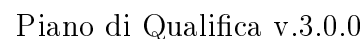
Test	Descrizione	Requisito
TS 61.4.10.10	Viene verificato che si possa visualizzare la riproduzione del video a schermo intero	RF 61.4.10.10
TS 61.7	Viene verificato che si possa passare da presentazione automatica a presentazione manuale e viceversa	RF 61.7, RF 61.10
TS 64	Viene verificato che si possa effettuare il Logout _g dal Server _g	RF 64
TS 67.1	Viene verificato che l'amministratore possa inserire dei Template _g di presentazioni	RF 67.1
TS 67.4	Viene verificato che l'amministratore possa inserire Template _g di Infografiche _g	RF 67.4
TS 67.7	Viene verificato che l'amministratore possa inserire elementi _g grafici	RF 67.7
TS 67.10	Viene verificato che l'amministratore possa eliminare un Template _g	RF 67.10
TS 67.13	Viene verificato che l'amministratore possa ripristinare un Template _g eliminato	RF 67.13
TS 70.1	Viene verificato che si possa selezionare una presentazione da cui produrre l'Infografica _g	RF 70.1
TS 70.4	Viene verificato che si possa selezionare Template _g di Infografica _g	RF 70.4
TS 70.5	Viene verificato che si possano selezionare gli elementi _g dell'Infografica _g	RF 70.5
TS 70.10	Viene verificato che si possa passare in modalità modifica di un'Infografica _g	RF 70.10
TS 70.10.1	Viene verificato che si possa modificare un Elemento _g di un'Infografica _g	RF 70.10.1

Università degli studi di Padova - 2014/2015

Test	Descrizione	Requisito
TS 70.10.19	Viene verificato che si possano eliminare elementi _g grafici o testuali	RF 70.10.19
TS 70.13	Viene verificato che si possa salvare l'Infografica _g 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 _g in formato stampabile	RF 70.19
TS 73	Viene verificato che si possa creare un'Infografica _g	RF 73

Tab 8: Descrizione dei test di sistema per i Requisiti di Qualità e Vincoli

Descrizione	Requisito
Viene verificato che ogni funzionalità dell'applicazione sia documentata	RQ _g 1, 7
Viene verificato che sia disponibile un tutorial per la creazione delle presentazioni	RQ _g 4
Viene verificato che sia disponibile la documentazione sui test eseguiti	RQ _g 10
Viene verificato che il sistema offra la possibilità di eseguire offline le presentazioni	RQ _g 13



Vengono descritti i test di integrazione che permettono di verificare la corretta integrazione e comunicazione tra parti distinte di sistema.

Di seguito viene riportato il diagramma per chiarire l'albero dei test d'integrazione.



Tab 9: Descrizione dei test di Integrazione

Test	Descrizione	Componenti aggiuntive	Stato
TIA1	Si verifica che NodeApi si integri correttamente con le classi che compongono AccessControll	NodeApi, AccessControll	N.E.
TIA2	Si verifica che il Loader interagisca correttamente con NodeApi e AccessControll	Loader	N.E.
TIA3	Si verifica che Registration interagisca correttamente con le NodeApi, Loader e AccessControll	Registration	N.E.
TIA4	Si verifica che Authentication interagisca correttamente con le NodeApi, Loader e AccessControll	Authentication	N.E.
TIC2	Si verifica che i componenti di SlideShowElements dialoghino correttamente con serverRelations		N.E.
TIC3	Si verifica che le classi che compongono InseritEditRemove interagiscano correttamente con SlideShowElements e serverRelations	SlideshowActions	N.E.
TIC4	Si verifica che le classi che compongono Command si integrino con InsertEditRemove		N.E.
TID1	Si verifica che il Controller ed il Model dialoghino correttamente	Controller	N.E.
TID2	Si verifica che la View si integri correttamente con il Controller	View	N.E.

B.3 Test d'unità

Tab 10: Descrizione dei test di Unità per il Server_g

Test	Descrizione	Metodi	Stato
TU1	viene verificato l'inserimento di un utente nel database in seguito ad una chiamata al metodo del Server_g con un utente non ancora registrato	POST Account/register	success
TU2	viene verificato che il Server_g non inserisce in database un utente già registrato chiamando il metodo opportuno del Server_g con un utente inserito	POST Account/register	success
TU3	viene verificato il metodo di autenticazione del Server_g attraverso una chiamata ad esso con un utente inserito e verificandone il token ritornato	GET Account/authenticate	success
TU4	viene verificato il metodo di autenticazione del Server_g attraverso una chiamata ad esso con un utente non inserito e verificando la risposta del Server_g	GET Account/authenticate	success
TU5	viene verificato il metodo di autenticazione del Server_g attraverso una chiamata ad esso con un utente inserito nel database ma con il parametro password errato e verificando la risposta del Server_g	GET Account/authenticate	success
TU6	viene verificato che il Server_g modifica la password salvata in database di un utente attraverso una chiamata con parametri corretti rispetto ad un utente inserito	POST Account/changedpassword	success
TU7	viene verificato che il Server_g non modifica la password salvata in database di un utente attraverso una chiamata con parametro un utente non inserito	POST Account/changedpassword	success



Test	Descrizione	Metodi	Stato
TU8	viene verificato che il Server _g non modifica la password salvata in database di un utente attraverso una chiamata con parametro password errato rispetto all'utente indicato	POST Account/changepassword	success
TU9	viene inserita una presentazione in database e viene verificato che una chiamata al metodo del Server _g ritorni un array il cui primo Elemento _g è corretto rispetto alla presentazione inserita	GET private/api/presentations	success
TU10	viene chiamato il metodo del Server _g per l'inserimento di una nuova presentazione, si verifica che la presentazione sia stata inserita e che abbia il nome corretto	POST private/api/presentations/new/[newpres]	success
TU11	viene chiamato il metodo del Server _g per l'inserimento di una nuova presentazione come copia di una già esistente e se ne verifica il corretto inserimento in database	POST private/api/presentations/new/[oldpres]/[newpres]	success
TU12	viene inserita una presentazione in database e viene chiamato il metodo del Server _g per il recupero della presentazione, si verifica il corretto recupero nella risposta	GET private/api/presentations/[presentation]	success
TU13	viene inserita una presentazione in database, poi viene chiamato il metodo del Server _g per la eliminazione della presentazione, si verifica la cancellazione della presentazione in database	DELETE private/api/presentations/[presentation]	success
TU14	viene inserita una presentazione in database e viene chiamato il metodo del Server _g per la rinomina della presentazione verificando che il nome sia cambiato con quello indicato	RENAME private/api/presentations/[oldname]/rename/[new]	success

Test	Descrizione	Metodi	Stato
TU15	viene inserito un Elemento _g in una presentazione in database, viene chiamato il metodo del Server _g con i parametri corretti e si osserva la cancellazione dell'Elemento _g dalla presentazione	DELETE private/api/presentation-s/[presname]/delete/[type]/[id]	success
TU16	viene chiamato il metodo del Server _g per l'inserimento di un Elemento _g in una presentazione, viene verificato l'inserimento dell'Elemento _g in database	POST private/api/presentation-s/[presname]/element	success
TU17	viene verificata la modifica dell'Elemento _g paths di una presentazione in database attraverso la chiamata al Server _g con dei parametri di esempio	PUT private/api/presentation-s/[presname]/paths	success
TU18	viene inserito un Elemento _g in una presentazione in database, viene chiamato il metodo del Server _g per la modifica di questo Elemento _g con dei valori di esempio, si verifica la modifica dei valori in database	PUT private/api/presentation-s/[presname]/element	success
TU19	viene inserita una immagine e viene chiamato il metodo del Server _g verificando che ritorni un'array con il primo Elemento _g avente il nome dell'immagine inserita	GET private/api/files/image	success
TU20	viene inserito un File _g immagine, viene chiamato il metodo del Server _g per la cancellazione di quel File _g e se ne verifica la cancellazione dalla opportuna cartella del Server _g	DELETE private/api/files/[image]	success
TU21	viene chiamato il metodo del Server _g per l'inserimento di un File _g immagine e si verifica l'inserimento del File _g nella cartella destinazione sul Server _g	POST private/api/files/[image]	success

Test	Descrizione	Metodi	Stato
TU22	viene inserito un File _g immagine, viene chiamato il metodo del Server _g per la rinomina del File _g , si verifica che il File _g nella cartella del Server _g sia stato rinominato	POST private/api/files/[image]/[newname]	success
TU23	viene inserita una immagine con valori degli attributi height e width noti, viene chiamato il metodo del Server _g per questa immagine e si verificano i valori ritornati con quelli noti	GET private/api/files/sizeImage/[image]	success
TU24	viene inserito un File _g audio e viene chiamato il metodo del Server _g verificando che ritorni un'array con il primo Elemento _g avente il nome del File _g inserito	GET private/api/files/audio	success
TU25	viene inserito un File _g audio, viene chiamato il metodo del Server _g per la cancellazione di quel File _g e se ne verifica la cancellazione dalla opportuna cartella del Server _g	DELETE private/api/files/[audio]	success
TU26	viene chiamato il metodo del Server _g per l'inserimento di un File _g audio e si verifica l'inserimento del File _g nella cartella destinazione sul Server _g	POST private/api/files/[audio]	success
TU27	viene inserito un File _g audio, viene chiamato il metodo del Server _g per la rinomina del File _g , si verifica che il File _g nella cartella del Server _g sia stato rinominato	POST private/api/files/[audio]/[newname]	success
TU28	viene inserito un File _g video e viene chiamato il metodo del Server _g verificando che ritorni un'array con il primo Elemento _g avente il nome del File _g inserito	GET private/api/files/video	success

Test	Descrizione	Metodi	Stato
TU29	viene inserito un File _g video, viene chiamato il metodo del Server _g per la cancellazione di quel File _g e se ne verifica la cancellazione dalla opportuna cartella del Server _g	DELETE private/api/files/[video]	success
TU30	viene chiamato il metodo del Server _g per l'inserimento di un File _g video e si verifica l'inserimento del File _g nella cartella destinazione sul Server _g	POST private/api/files/[video]	success
TU31	viene inserito un File _g video, viene chiamato il metodo del Server _g per la rinomina del File _g , si verifica che il File _g nella cartella del Server _g sia stato rinominato	POST private/api/files/[video]/[newname]	success

Tab 11: Descrizione dei test di Unità per ServerRelation; classe: fileServerRelation

Test	Descrizione	Metodi	Stato
TU32	viene verificato che il ritorno della Funzione _g sia corrispondente ai File _g immagini dell'utente nel Server _g	getMetasImages()	success
TU33	viene verificato che il ritorno della Funzione _g sia corrispondente ai File _g audio dell'utente nel Server _g	getMetasAudios()	success
TU34	viene verificato che il ritorno della Funzione _g sia corrispondente ai File _g video dell'utente nel Server _g	getMetasVideos()	success
TU35	viene verificato che il ritorno della Funzione _g sia corretto rispetto ai valori di width e height del File _g immagine inserito nel Server _g	getImageDimension(img:string)	success
TU36	viene verificata l'eliminazione di un File _g immagine con la chiamata della Funzione _g su un File _g immagine inserito nel Server _g	deleteImage(img:string)	success

Test	Descrizione	Metodi	Stato
TU37	viene verificata l'eliminazione di un File _g audio con la chiamata della Funzione _g su un File _g audio inserito nel Server _g	deleteAudio(img:string)	success
TU38	viene verificata l'eliminazione di un File _g video con la chiamata della Funzione _g su un File _g video inserito nel Server _g	deleteVideo(img:string)	success
TU39	viene verificata la rinominazione di un File _g immagine inserito nel Server _g attraverso la chiamata alla Funzione _g con i parametri corretti	renameImage(old:string,new:string)	success
TU40	viene verificata la rinominazione di un File _g audio inserito nel Server _g attraverso la chiamata alla Funzione _g con i parametri corretti	renameAudio(old:string,new:string)	success
TU41	viene verificata la rinominazione di un File _g video inserito nel Server _g attraverso la chiamata alla Funzione _g con i parametri corretti	renameVideo(old:string,new:string)	success

Tab 12: Descrizione dei test di Unità per ServerRelation; classe: serverRelation

Test	Descrizione	Metodi	Stato
TU42	viene inserita una presentazione in database e viene verificato che il ritorno della Funzione _g sia consistente che le presentazioni presenti in database	getPresentationsMeta()	success
TU43	viene chiamata la Funzione _g per l'inserimento di una nuova presentazione, viene verificato l'inserimento della presentazione in database	newPresentation(name:string)	success

Test	Descrizione	Metodi	Stato
TU44	viene chiamata la Funzione _g per l'inserimento di una nuova presentazione come copia di una già esistente, viene verificato l'inserimento della presentazione in database come copia della esistente	newCopyPresentation(name:string)	success
TU45	viene inserita una presentazione in database e viene chiamato il metodo del Server _g per il recupero della presentazione, si verifica il corretto ritorno della Funzione _g	getPresentation(name:string)	success
TU46	viene inserita una presentazione in database e viene chiamato il metodo di eliminazione della presentazione dell'utente, si verifica la cancellazione della presentazione	deletePresentation(name:string)	success
TU47	viene inserita una presentazione in database e viene chiamato il metodo di rinominazione della presentazione dell'utente, si verifica la rinominazione della presentazione	renamePresentation(old:string,new:string)	success
TU48	viene chiamato il metodo del Server _g per l'inserimento di un Elemento _g in una presentazione, viene verificato l'inserimento dell'Elemento _g in database	newElement(presentation:string , onj:object , callback:function)	success
TU49	viene inserito un Elemento _g in una presentazione in database, viene chiamato il metodo con i parametri corretti e si osserva la modifica dell'Elemento _g dalla presentazione	updateElement(presentation :string , onj : object , callback : function)	success
TU50	viene inserito un Elemento _g in una presentazione in database, viene chiamato il metodo con i parametri corretti e si osserva la cancellazione dell'Elemento _g dalla presentazione	deleteElement(presentation : string, type : string , id : string)	success

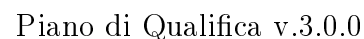
Test	Descrizione	Metodi	Stato
TU51	viene chiamato il metodo per la modifica del campo paths di una presentazione e si osserva la modifica del campo dati della presentazioni in database	updatePaths(presentation:string,obj:object)	success

Tab 13: Descrizione dei test di Unità per ServerRelation; classe: Loader

Test	Descrizione	Metodi	Stato
TU52	vengono chiamati i metodi addInsert(), addUpdate e addDelete() su un oggetto mook per la simulazione di altre classi del model per il recupero degli elementi della presentazione associati, si verifica lo stato della presentazione alle diverse chiamate del metodo	update()	success

Tab 14: Descrizione dei test di Unità

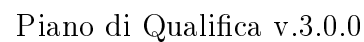
Test	Descrizione	Stato
TU53	Si verifica che le classi Services vengano istanziate correttamente	Success
TU54	Si verifica che HeaderComponent venga istanziato correttamente	Success
TU55	Si verifica che Authentication-Controller venga istanziato correttamente	Success
TU56	Si verifica che HomeController venga istanziato correttamente	Success
TU57	Si verifica che ProfileController venga istanziato correttamente	Success
TU58	Si verifica che ExecutionController venga istanziato correttamente	Success
TU59	Si verifica che EditController venga istanziato correttamente	Success



Test	Descrizione	Metodi	Stato
TU60	Richiama il costruttore di Text(...) e verifica che sia stato creato un oggetto di tipo text	slideShowElements::Text()	success
TU61	Richiama il costruttore di Frame(...) e verifica che sia stato creato un oggetto di tipo Frame _g	slideShowElements::Frame()	success
TU62	Richiama il costruttore di Image(...) e verifica che sia stato creato un oggetto di tipo image	slideShowElements::Image()	success
TU63	Richiama il costruttore di Audio(...) e verifica che sia stato creato un oggetto di tipo audio	slideShowElements::Audio()	success
TU64	Richiama il costruttore di Video(...) e verifica che sia stato creato un oggetto di tipo video	slideShowElements::Video()	success
TU65	Richiama il costruttore di Background(...) e verifica che sia stato creato un oggetto di tipo background	slideShowElements::Background()	success

Test	Descrizione	Metodi	Stato
TU66	Richiama il costruttore di InsertEditRemove() e verifica che venga istanziato un oggetto di tale classe	slideShowActions::InsertEditRemove()	success
TU67	Richiama il costruttore di InsertEditRemove() e verifica che sia stata generata una sola istanza di tale classe	slideShowActions::InsertEditRemove()	success
TU68	Richiama il InsertEditRemove()::insertText e verifica che venga inserito nella presentazione un oggetto Text	slideShowActions:: InsertEditRemove():: insertText(...)	success

Test	Descrizione	Metodi	Stato
TU77	Richiama il InsertEditRemove()::editBackground(...) su ogni Elemento _g della presentazione e verifica che ogni Elemento _g venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::editBackground(...)	success
TU78	Richiama il InsertEditRemove()::editColor(...) su ogni Elemento _g della presentazione e verifica che ogni Elemento _g venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::editColor(...)	success
TU79	Richiama il InsertEditRemove()::editShape(...) su ogni Elemento _g della presentazione e verifica che ogni Elemento _g venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::editPosition(...)	success
TU80	Richiama il InsertEditRemove()::removeText(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeText(...)	success
TU81	Richiama il InsertEditRemove()::removeFrame(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeFrame(...)	success
TU82	Richiama il InsertEditRemove()::removeImage(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeImage(...)	success
TU83	Richiama il InsertEditRemove()::removeAudio(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeAudio(...)	success
TU84	Richiama il InsertEditRemove()::removeVideo(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeVideo(...)	success

Università degli studi di Padova - 2014/2015