

08-08-2015



# Piano di Qualifica

## Informazioni sul documento

<b>Nome Documento</b>	Piano di Qualifica
<b>Versione</b>	4.0.0
<b>Stato</b>	<i>Formale</i>
<b>Uso</b>	<i>Esterno</i>
<b>Data Creazione</b>	20-03-2015
<b>Data Ultima Modifica</b>	08-08-2015
<b>Redazione</b>	Petrucci Mauro
<b>Approvazione</b>	Fossa Manuel
<b>Verifica</b>	Tollot Pietro
<b>Lista distribuzione</b>	<i>LateButSafe</i>

Prof. Tullio Vardanega

Prof. Riccardo Cardin

Proponente Zucchetti S.p.a.



Tab 1: Versionamento del documento

Università degli studi di Padova - 2014/2015

Versione	Autore	Data	Descrizione
0.1.0	Busetto Matteo	20-03-2015	Stesura dello scheletro del documento



## pre-RR

Tab 2: Storico ruoli pre-RR

Tab 3: Storico ruoli RR  $\rightarrow$  RP

Tab 4: Storico ruoli RP  $\rightarrow$  RQ

---

Università degli studi di Padova - 2014/2015

<b>Versione 4.0.0</b>	<b>Nominativo</b>
Redazione	Petrucci Mauro
Verifica	Tollot Pietro
Approvazione	Fossa Manuel

Tab 5: Storico ruoli RQ  $\rightarrow$  RA

# Indice

<b>1</b>	<b>Introduzione</b>	<b>8</b>
1.1	Scopo del documento . . . . .	8
1.2	Scopo del Prodotto . . . . .	8
1.3	Glossario . . . . .	8
1.4	Riferimenti . . . . .	8
1.4.1	Normativi . . . . .	8
1.4.2	Informativi . . . . .	8
<b>2</b>	<b>Obiettivi di qualità</b>	<b>10</b>
2.1	Qualità di processo . . . . .	10
2.2	Qualità di prodotto . . . . .	12
2.2.1	Funzionalità . . . . .	12
2.2.2	Affidabilità . . . . .	13
2.2.3	Efficienza . . . . .	13
2.2.4	Usabilità . . . . .	13
2.2.5	Manutenibilità . . . . .	13
2.2.6	Portabilità . . . . .	14
2.3	Procedure di controllo di qualità di processo . . . . .	14
<b>3</b>	<b>Visione generale delle strategie di verifica</b>	<b>16</b>
3.1	Organizzazione . . . . .	16
3.2	Pianificazione strategica e temporale . . . . .	16
3.3	Responsabilità . . . . .	17
3.4	Risorse . . . . .	17
3.5	Tecniche di analisi . . . . .	18
3.6	Metriche . . . . .	19
3.6.1	metriche <sub>g</sub> per la progettazione . . . . .	19
3.6.2	metriche <sub>g</sub> per il codice . . . . .	19
3.6.3	metriche <sub>g</sub> per i documenti . . . . .	21
<b>Appendice A PDCA</b>		<b>22</b>
A.1	Revisione di accettazione . . . . .	22

<b>Appendice B Riassunto delle attività di verifica</b>	<b>23</b>
B.1 Revisione dei Requisiti . . . . .	23
B.2 Documenti . . . . .	23
B.3 Progettazione Architettuale . . . . .	23
B.4 Progettazione di dettaglio e codifica . . . . .	24
B.5 Miglioramenti post Revisione . . . . .	25
B.5.1 Miglioramenti post revisione di Qualifica . . . . .	25
<b>Appendice C Pianificazione dei test</b>	<b>27</b>
C.1 Test di sistema . . . . .	27
C.2 Test d'integrazione . . . . .	37
C.3 Test d'unità . . . . .	39

## Elenco delle figure

1	Rappresentazione del modello ISO/IEC 9126:2001 . . . . .	12
2	Schema PDCA . . . . .	14
3	Sequenza d'integrazione delle componenti . . . . .	37

## Elenco delle tabelle

1	Versionamento del documento . . . . .	1
2	Storico ruoli pre-RR . . . . .	3
3	Storico ruoli RR -> RP . . . . .	3
4	Storico ruoli RP -> RQ . . . . .	3
5	Esiti verifica documenti, Analisi . . . . .	23
6	Tabella accoppiamento afferente ed efferente delle componenti . . . . .	24
7	Problemi individuati in RQ e relative soluzioni. . . . .	26
8	Descrizione dei test di sistema per i Requisiti <sub>g</sub> Funzionali . . . . .	27
9	Descrizione dei test di sistema per i Requisiti <sub>g</sub> di Qualità e Vincoli . . . . .	36
10	Descrizione dei test di Integrazione . . . . .	38
11	Descrizione dei test di Unità per il Server <sub>g</sub> . . . . .	39
12	Descrizione dei test di Unità per ServerRelation; classe: fileServerRelation . . .	43
13	Descrizione dei test di Unità per ServerRelation; classe: serverRelation . . . .	44
14	Descrizione dei test di Unità per ServerRelation; classe: Loader . . . . .	46
15	Descrizione dei test di Unità . . . . .	46
16	Descrizione dei test di Unità per SlideShowElements . . . . .	47
17	Descrizione dei test di Unità per InsertEditRemove . . . . .	47

# 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<sub>g</sub> di verifica continua sulle attività svolte; questo consentirà di rilevare e correggere anomalie e incongruenze in modo tempestivo e senza spreco di Risorse<sub>g</sub>.

## 1.2 Scopo del Prodotto

Lo scopo del Progetto<sub>g</sub> è la realizzazione un Software<sub>g</sub> 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<sub>g</sub>: [NormeDiProgetto\\_v.4.0.0.pdf](#);
- Capitolato d'appalto C4: Premi: Software<sub>g</sub> di presentazione “better than Prezi”  
<http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf>.

### 1.4.2 Informativi

- Piano di Progetto<sub>g</sub>: [PianoDiProgetto\\_v.4.0.0.pdf](#);
- Slide dell'insegnamento Ingegneria del Software<sub>g</sub> modulo A:  
<http://www.math.unipd.it/~tullio/IS-1/2014/> ;
- SWEBOK – Version 3 (2004): capitolo 11 – Software<sub>g</sub> Quality  
<http://www.computer.org/portal/web/swebok/html/ch11>;
- Ingegneria del Software<sub>g</sub> - Ian Sommerville - 9a Edizione (2010):
  - Capitolo 24 - Gestione della qualità;
  - Capitolo 26 - Miglioramento dei Processi<sub>g</sub>.
- Standard ISO<sub>g</sub> /IEC TR 15504: Software<sub>g</sub> process assessment  
[http://en.wikipedia.org/wiki/ISO/IEC\\_15504](http://en.wikipedia.org/wiki/ISO/IEC_15504);

- Standard ISO<sub>g</sub> /IEC 9126: Product quality  
[http://en.wikipedia.org/wiki/ISO/IEC\\_9126](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<sub>g</sub> che lo definiscono; per questo motivo si è deciso di utilizzare lo standard ISO/IEC 15504 denominato SPICE<sub>g</sub>, che rende disponibili strumenti adatti a valutarli.

Per applicare correttamente questo modello si deve utilizzare il ciclo di Deming (ciclo PDCA) il quale definisce una metodologia di controllo dei Processi<sub>g</sub> durante il loro ciclo di vita<sub>g</sub> che consente di migliorarne in modo continuativo la qualità.

Tutti i Processi<sub>g</sub> dovranno quindi essere sottoposti a valutazione in modo da verificarne la qualità ed eventualmente facilitarne il miglioramento. A tale scopo lo SPICE<sub>g</sub> definisce nove attributi di Processo<sub>g</sub> per effettuare una migliore valutazione:

## 1. Process performance

Gli indicatori della performance di  $\text{Processo}_g$  sono:

- I lavori identificati come input al Processo<sub>g</sub> (input work products);
- I lavori identificati come output del Processo<sub>g</sub> (output work products);
- Le azioni compiute per trasformare gli input work products in output work products.

## 2. Performance Management

L'attuazione di un Processo<sub>g</sub> è pianificata e controllata al fine di generare risultati che rispondono agli obiettivi attesi;

### 3. Work Product Management

L'attuazione di un Processo<sub>g</sub> è pianificata e controllata al fine di generare risultati che siano adeguatamente documentati, controllati e verificati;

## 4. Process Definition

L'attuazione di un Processo<sub>g</sub> si basa su approcci standardizzati;

## 5. Process Resource

Il Processo<sub>g</sub> può contare su adeguate risorse umane<sub>g</sub>, di infrastrutture, ecc. per essere attuato;

## 6. Process Measurement

I risultati conseguiti e le misure rilevate durante l'attuazione di un Processo<sub>g</sub> sono utilizzati per assicurarsi che l'attuazione di tale Processo<sub>g</sub> supporti efficacemente il raggiungimento di obiettivi specifici;

## 7. Process Control

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

## 8. Process Change

Le modifiche alla definizione, gestione e attuazione di un Processo<sub>g</sub> sono controllate;

## 9. Continuous Integration

Le modifiche ad un Processo<sub>g</sub> sono identificate ed implementate con lo scopo di assicurare il continuo miglioramento nel raggiungere gli obbiettivi definiti per l'organizzazione.

Sono inoltre stabiliti quattro differenti livelli di possesso di ciascuno degli attributi:

- **N - Non posseduto** (0 - 15% di possesso): non c'è evidenza oppure ce n'è poca del possesso di un attributo;
- **P - Parzialmente posseduto** (16 - 50% di possesso): c'è evidenza di approccio sistematico al raggiungimento del possesso di un attributo e del raggiungimento di tale possesso, ma alcuni aspetti del possesso possono essere non prevedibili;
- **L - Largamente posseduto** (51 - 85% di possesso): vi è evidenza di approccio sistematico al raggiungimento del possesso di un attributo e di un significativo livello di possesso di tale attributo, ma l'attuazione del Processo<sub>g</sub> può variare nelle diverse unità operative dell'organizzazione;
- **F - (Fully) Pienamente posseduto** (86 - 100% di possesso): vi è evidenza di un totale e sistematico approccio e di un completo raggiungimento del possesso dell'attributo; non esistono significative differenze nel modo di attuare il Processo<sub>g</sub> tra le diverse unità operative.

Vi sono poi vari livelli di maturità dei Processi<sub>g</sub> che dipendono dal diverso livello di possesso degli attributi:

- **Livello 0** - Processo<sub>g</sub> incompleto: il Processo<sub>g</sub> non è implementato o non raggiunge gli obiettivi. Non vi è evidenza di approcci sistematici agli attributi definiti;
- **Livello 1** - Processo<sub>g</sub> semplicemente attuato: il Processo<sub>g</sub> viene messo in atto e raggiunge i suoi obiettivi. Non vi è evidenza di approcci sistematici agli attributi definiti. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process performance”;
- **Livello 2** - Processo<sub>g</sub> gestito: il Processo<sub>g</sub> è attuato, ma anche pianificato, tracciato, verificato ed aggiustato se necessario, sulla base di obiettivi ben definiti. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Performance management” e “Work product management”;
- **Livello 3** - Processo<sub>g</sub> definito: il Processo<sub>g</sub> è attuato, pianificato e controllato sulla base di procedure ben definite, basate sui principi del Software<sub>g</sub> engineering. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process definition” e “Process resource” ;
- **Livello 4** - Processo<sub>g</sub> predicibile: il Processo<sub>g</sub> è stabilizzato ed è attuato all’interno di definiti limiti riguardo i risultati attesi, le performance, le Risorse<sub>g</sub> impiegate ecc. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process measurement” e “Process control”;

- **Livello 5** - Processo<sub>g</sub> ottimizzante: il Processo<sub>g</sub> 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<sub>g</sub> che ai suoi utilizzatori o acquirenti. Per gli sviluppatori porta vantaggi nell'ottimizzazione dell'uso delle Risorse<sub>g</sub>, 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<sub>g</sub>, 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<sub>g</sub>, individuando sei caratteristiche principali atte a rendere il prodotto qualitativamente accettabile.

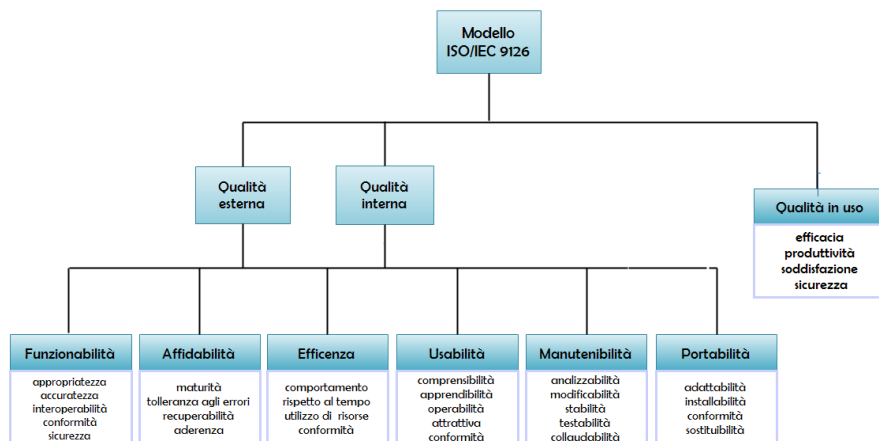


Fig 1: Rappresentazione del modello ISO/IEC 9126:2001

### 2.2.1 Funzionalità

È un Requisito<sub>g</sub> funzionale che indica la capacità del Software<sub>g</sub> di soddisfare le esigenze espresse dal capitolato ed individuate durante l'analisi dei Requisiti<sub>g</sub>. Per valutare questa caratteristica si considerano l'appropriatezza e l'accuratezza delle funzioni<sub>g</sub> 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<sub>g</sub> soddisfatti che

avranno un riscontro in elementi<sub>g</sub> funzionanti nell'applicazione prodotta. La soglia di sufficienza sarà quindi data dal soddisfacimento di tutti i Requisiti<sub>g</sub> obbligatori previsti dal capitolato d'appalto.

### 2.2.2 Affidabilità

È un Requisito<sub>g</sub> non funzionale che indica la capacità del Software<sub>g</sub> 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<sub>g</sub> non funzionale che indica il rapporto tra le prestazioni e le Risorse<sub>g</sub> disponibili. Si valuta se il Software<sub>g</sub> utilizza al meglio le Risorse<sub>g</sub> 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<sub>g</sub>, data dalla capacità d'utilizzarne una quantità adeguata ad eseguire le funzioni<sub>g</sub> richieste.

**Quantificazione del raggiungimento dell'obiettivo di qualità:** Un modo per valutare l'efficienza di un Software<sub>g</sub> 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<sub>g</sub> non funzionale che indica la capacità del Software<sub>g</sub> 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<sub>g</sub> per quantificarla, perciò si farà affidamento alle linee guida del material design<sub>g</sub> fornite da Google, dato l'alto tasso di adozione rispetto ad altre linee guida.

### 2.2.5 Manutenibilità

È un Requisito<sub>g</sub> non funzionale che indica la capacità del Software<sub>g</sub> 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<sub>g</sub> si fa riferimento alle metriche<sub>g</sub> descritte nella sezione 3.6.

### 2.2.6 Portabilità

È un Requisito<sub>g</sub> non funzionale che indica la capacità del Software<sub>g</sub> 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<sub>g</sub> funzioni<sub>g</sub> sia su computer (indipendentemente dal loro sistema operativo) e su dispositivi mobile<sub>g</sub> Android<sub>g</sub>, iOS e Windows<sub>g</sub> Phone.

## 2.3 Procedure di controllo di qualità di processo

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

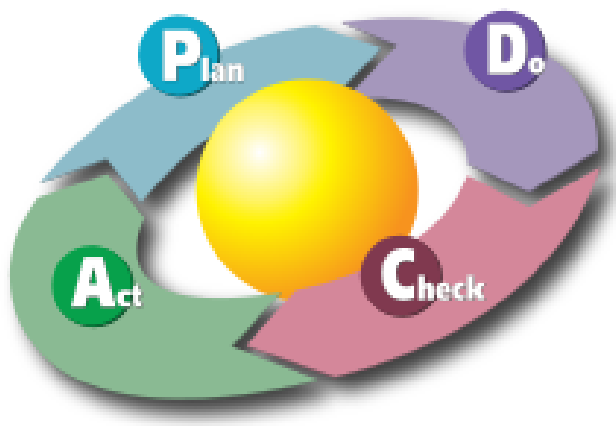
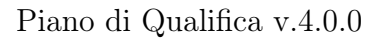


Fig 2: Schema PDCA

È chiamato anche ciclo PDCA, in quanto è definito dall'Iterazione<sub>g</sub> delle quattro fasi:

- **Plan:** si stabiliscono obiettivi e Processi<sub>g</sub> necessari ad ottenere risultati conformi agli obiettivi attesi;
- **Do:** si implementa il piano, si esegue il Processo<sub>g</sub> 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.4.0.0*(§5.1).





## 3 Visione generale delle strategie di verifica

### 3.1 Organizzazione

Ogniquale volta avvenga un cambiamento sostanziale nello sviluppo del prodotto, si istanzierà il Processo<sub>g</sub> di verifica.

Nello specifico durante ogni fase (Analisi, Progettazione, Realizzazione e Validazione<sub>g</sub>) 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<sub>g</sub> di Codice<sub>g</sub>;
- 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.4.0.0.pdf](#);
- Conclusione della modifica sostanziale di un File<sub>g</sub> di Codice<sub>g</sub>, 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.4.0.0.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<sub>g</sub> 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<sub>g</sub>.

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 erroneo ma non vengono lette, o non viene Percorso<sub>g</sub> il ramo di Codice<sub>g</sub> 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<sub>g</sub> ed evitare la propagazione di errori all'interno dei documenti o di File<sub>g</sub> di Codice<sub>g</sub> 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



la correzione di imprecisioni concettuali e tecniche potrebbe risultare particolarmente gravosa. Come da Piano di Progetto<sub>g</sub> di seguito si riportano le quattro Milestone<sub>g</sub> prima delle quali si effettuerà una verifica del prodotto:

- Revisioni formali:
  - Revisione dei Requisiti<sub>g</sub> (28/04/2015)
  - Revisione di Accettazione (06/07/2015)
- Revisioni di progresso:
  - Revisione di Progettazione (29/05/2015)
  - Revisione di Qualifica (18/06/2015)

Sarà necessario, infine, assicurarsi che ogni Requisito<sub>g</sub> sia tracciato consistentemente nel documento di Analisi dei Requisiti<sub>g</sub>.

### 3.3 Responsabilità

I principali ruoli di responsabilità individuati sono:

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

### 3.4 Risorse

Per assicurare che gli obiettivi qualitativi vengano raggiunti è necessario l'utilizzo di Risorse<sub>g</sub> sia umane che tecnologiche. Per una dettagliata descrizione dei ruoli e delle loro responsabilità fare riferimento alle Norme di Progetto<sub>g</sub>. Per Risorse<sub>g</sub> tecniche e tecnologiche sono da intendersi tutti gli strumenti Software<sub>g</sub> e hardware che il gruppo intende utilizzare per attuare le attività di verifica su Processi<sub>g</sub> 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<sub>g</sub>.

### 3.5 Tecniche di analisi

- **Analisi statica:** consiste nell'analizzare il Codice<sub>g</sub> 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<sub>g</sub>, 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<sub>g</sub> che sono ritenute fonti maggiori di errore. La *lista di controllo*, che deve essere seguita per svolgere efficacemente questo Processo<sub>g</sub>, 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.4.0.0*.
- **Analisi dinamica:** consiste nel verificare e validare il Software<sub>g</sub> 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<sub>g</sub>, generalmente prodotte da un singolo programmatore, in modo da verificare che rispettino i Requisiti<sub>g</sub>. 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<sub>g</sub> Software<sub>g</sub> individuati nell'analisi dei Requisiti<sub>g</sub> permettendo la Validazione<sub>g</sub> del sistema prodotto;
  - **Test di regressione:** stabilisce se modifiche all'implementazione di un Programma<sub>g</sub> alterano elementi<sub>g</sub> 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<sub>g</sub> tra utente e sistema;
- **Test di collaudo:** attività formale supervisionata dal Committente<sub>g</sub> il cui buon esito comporta la possibilità di rilasciare il prodotto;

## 3.6 Metriche

Il Processo<sub>g</sub> di verifica, per essere informativo, deve essere quantificabile. Le misure rilevate dal Processo<sub>g</sub> di verifica devono quindi essere basate su metriche<sub>g</sub> stabilite a priori.

Una Metrica<sub>g</sub> è la misura di una proprietà relativa ad una porzione di un documento Software<sub>g</sub>, allo scopo di fornire informazioni significative sulla qualità del Codice<sub>g</sub> prodotto.

Per valutare la bontà del lavoro svolto non è sufficiente basarsi solo sulle metriche<sub>g</sub>, che sono solamente degli indicatori valutati a posteriori, perché un'importanza ancora maggiore la riveste il controllo sulla qualità del Processo<sub>g</sub>.

### 3.6.1 metriche<sub>g</sub> 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<sub>g</sub> (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:

$$(\text{lunghezzafunzione})^2 \times \text{fan} - \text{in} \times \text{fan} - \text{out}.$$

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<sub>g</sub>.

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<sub>g</sub> per il codice

- **Complessità Ciclomantica di McCabe:** è indicazione del numero di segmenti lineari in un metodo (ad esempio sezioni di Codice<sub>g</sub> 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 Ciclomantica 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<sub>g</sub> semplice con pochi rischi, superato questo limite il Codice<sub>g</sub> diventa più complesso, instabile e difficilmente manutenibile;

- **Numero linee di codice:** rappresenta il numero di linee di Codice<sub>g</sub> all'interno di un blocco. Un indice elevato non rappresenta necessariamente un cattivo Codice<sub>g</sub> ma suggerisce la possibilità di estrarre metodi contenenti gruppi di istruzioni correlate, aumentando il livello di astrazione;
- **Halstead:** la Metrica<sub>g</sub> di Halstead<sub>g</sub> non è solamente un indice di complessità, ma identifica le proprietà misurabili del Software<sub>g</sub> e le relative relazioni. Si basa sull'osservazione che una Metrica<sub>g</sub> 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<sub>g</sub> 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 `Funzioneg`. Il volume di una function senza parametri composta da una sola linea è 20, mentre un indice superiore a 1000 indica che probabilmente la `Funzioneg` esegue troppe operazioni.

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

### Parametri utilizzati

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

- **Indice di manutenibilità:** Questa Metrica<sub>g</sub> è una scala logaritmica con valore massimo 171. Rappresenta quanto manutenibile è il Codice<sub>g</sub>, ossia quanto facile è da supportare e migliorare.

L'indice di manutenibilità è calcolato tramite una fattorizzazione di altre metriche<sub>g</sub> 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<sub>g</sub>:

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

- **Copertura del codice:** è indicazione di quanto Codice<sub>g</sub> sorgente sia stato testato. Un elevato indice di copertura indica che il Codice<sub>g</sub> 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<sub>g</sub> 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 PDCA

## A.1 Revisione di accettazione

- **PLAN** La revisione di accettazione prevede la chiusura della documentazione e la consegna del relativo software prodotto. Il software va testato approfonditamente in modo scrupoloso. La qualità del software dovrà migliorare andando ad intervenire laddove gli indici metrici segnalino valori degni di attenzione. Il collaudo verificherà le funzionalità del sistema, permetterà di scovare e segnalare buona parte delle anomalie, in questo modo si arriverà in sede di revisione di accettazione con un prodotto sufficientemente pronto ad una dimostrazione.
- **CHECK** La validazione ha evidenziato alcune anomalie nel software, alcune segnalate dal proponente stesso. La correzione ha riguardato per lo più piccole funzionalità software.



## B.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.4.0.0* e alcune mancanze all'interno del documento di *Analisi dei Requisiti v.4.0.0*.

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<sub>g</sub> descritte su §3.6.3.

Tab 6: 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.

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 7: 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.

## B.4 Progettazione di dettaglio e codifica

Vengono qui riportate le misure rilevate con le metriche<sub>g</sub> sull'analisi statica<sub>g</sub> e dinamica del Codice<sub>g</sub>. Per ogni Metrica<sub>g</sub> si riportano i valore calcolati mantenendo una separazione tra back end e front-end. Per una descrizione delle metriche<sub>g</sub> 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<sub>g</sub> rispettano i parametri ottimali prefissati al contrario il Front-end ha una complessità ciclomatica troppo elevata.

### Halstead<sub>g</sub> Volume

- **Back-end**



- Medio: 53.3
- Massimo: 228

- **Front-end**

- Medio: 326.6
- Massimo: 1304

I Valori sia del Back-end<sub>g</sub> 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<sub>g</sub> 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<sub>g</sub> questo a causa della difficoltà nel testare i controller ed ottenere una copertura elevata.

Il Back-end<sub>g</sub> rientra nei parametri ottimali prefissati.

## B.5 Miglioramenti post Revisione

### B.5.1 Miglioramenti post revisione di Qualifica

A seguito delle attività di verifica e controllo si è cercato di identificare le problematiche relative ai processi e formulare proposte risolutive. Da queste idee sono nate diverse modifiche e miglioramenti ai documenti e in generale al nostro modo di lavorare. Seguendo questa linea abbiamo applicato coerentemente la politica di plan-do-check-act



Tab 8: Problemi individuati in RQ e relative soluzioni.

## C Pianificazione dei test

Si descrivono di seguito tutti i test di Validazione<sub>g</sub>, 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.4.0.0.pdf](#).

## C.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<sub>g</sub> descritti nell' [AnalisiDeiRequisiti\\_v.4.0.0.pdf](#). I test di sistema riportati sono quelli relativi ai Requisiti<sub>g</sub> Software<sub>g</sub> individuati e pertanto meritevoli di un test.

Tab 9: 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 <sub>g</sub>	RF 7
TS 7.1	Viene verificato che si possa inserire un nuovo Frame <sub>g</sub> nel piano della presentazione <sub>g</sub>	RF 7.1

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

Test	Descrizione	Requisito
TS 7.7.34	Viene verificato che si possa modificare la dimensione di un Frame <sub>g</sub>	RF 7.7.34
TS 7.7.37	Viene verificato che si possa modificare lo spessore del bordo di un Frame <sub>g</sub>	RF 7.7.37
TS 7.7.40	Viene verificato che si possa modificare il colore del bordo di un Frame <sub>g</sub>	RF 7.7.40
TS 7.7.43	Viene verificato che si possa modificare lo sfondo di un Frame <sub>g</sub>	RF 7.7.43
TS 7.7.46	Viene verificato che si possa ruotare un Frame <sub>g</sub>	RF 7.7.46
TS 7.10	Viene verificato che si possa eliminare un Frame <sub>g</sub> 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 <sub>g</sub> di visualizzazione	RF 7.19
TS 7.19.1	Viene verificato che si possa impostare un Frame <sub>g</sub> iniziale per il Percorso <sub>g</sub> di presentazione	RF7.19.1
TS 7.19.4	Viene verificato che si possa definire una transizione tra due Frame <sub>g</sub>	RF 7.19.4
TS 7.19.10	Viene verificato che si possa eliminare una transizione tra due Frame <sub>g</sub>	RF 7.19.10
TS 7.19.13	Viene verificato che si possa togliere un Frame <sub>g</sub> dal Percorso <sub>g</sub> di presentazione	RF 7.19.13

Test	Descrizione	Requisito
TS 7.22	Viene verificato che si possa assegnare un Bookmark <sub>g</sub> ad un Frame <sub>g</sub>	RF 7.22
TS 7.25	Viene verificato che si possa rimuovere un Bookmark <sub>g</sub> da un Frame <sub>g</sub>	RF 7.25
TS 7.28	Viene verificato che si possa modificare la velocità di transizione tra due Frame <sub>g</sub> consecutivi	RF 7.28
TS 7.31	Viene verificato che si possa impostare un effetto di transizione tra due Frame <sub>g</sub> consecutivi	RF 7.31
TS 7.34	Viene verificato che si possa impostare il tempo di attesa tra due Frame <sub>g</sub> consecutivi durante la riproduzione automatica	RF 7.34
TS 7.37	Viene verificato che si possa inserire un Elemento <sub>g</sub> SVG in un Frame <sub>g</sub> o nel piano della presentazione <sub>g</sub>	RF 7.37
TS 7.40.1	Viene verificato che si possa modificare le dimensioni di un Elemento <sub>g</sub> SVG	RF 7.40.1
TS 7.40.2	Viene verificato che si possa modificare il colore di un Elemento <sub>g</sub> SVG	RF 7.40.2
TS 7.43	Viene verificato che si possa eliminare un Elemento <sub>g</sub>	RF 7.43
TS 7.46	Viene verificato che si possa ruotare un Elemento <sub>g</sub>	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 <sub>g</sub>	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 <sub>g</sub>	RF 10.4
TS 10.5	Viene verificato che si possa assegnare un Bookmark <sub>g</sub> ad un Frame <sub>g</sub> da mobile	RF 10.5
TS 10.8	Viene verificato che si possa rimuovere un Bookmark <sub>g</sub> ad un Frame <sub>g</sub> da mobile	RF 10.8
TS 13	Viene verificato che si possa caricare un File <sub>g</sub> media presente in locale nel Server <sub>g</sub>	RF 13
TS 16	Viene verificato che si possano eliminare dal Server <sub>g</sub> i File <sub>g</sub> media caricati	RF 16
TS 17	Viene verificato che si possano rinominare i File <sub>g</sub> media presenti sul Server <sub>g</sub>	RF 17
TS 19	Viene verificato che si possano rinominare le presentazioni salvate nel Server <sub>g</sub>	RF 19
TS 25	Viene verificato che si possano rinominare le Infografiche <sub>g</sub> salvate nel Server <sub>g</sub>	RF 25
TS 31	Viene verificato che si possa eliminare dal Server <sub>g</sub> un'Infografica <sub>g</sub> creata	RF 31
TS 34	Viene verificato che si possa eliminare dal Server <sub>g</sub> una presentazione creata	RF 34
TS 35	Viene verificato che si possano visualizzare le presentazioni salvate sul Server <sub>g</sub>	RF 35
TS 36	Viene verificato che si possano visualizzare le Infografiche <sub>g</sub> salvate sul Server <sub>g</sub>	RF 36



Test	Descrizione	Requisito
TS 37	Viene verificato che si possano visualizzare i File <sub>g</sub> media salvati sul Server <sub>g</sub>	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 <sub>g</sub> creata sul Server <sub>g</sub>	RF 46
TS 49	Viene verificato che si possa salvare in locale una presentazione creata sul Server <sub>g</sub>	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 <sub>g</sub> successivo o al precedente	RF 61.1.1
TS 61.1.4	Viene verificato che si possa selezionare un Elemento scelta <sub>g</sub> se presente nel Frame <sub>g</sub>	RF 61.1.4
TS 61.1.7	Viene verificato che si possa passare al Frame <sub>g</sub> con Bookmark <sub>g</sub> successivo o precedente	RF 61.1.7
TS 61.1.10	Viene verificato che si possa passare da un Frame <sub>g</sub> visualizzato al suo Frame <sub>g</sub> contenitore	RF 61.1.10
TS 61.1.13	Viene verificato che si possa zoomare in una parte qualsiasi del Frame <sub>g</sub>	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 <sub>g</sub> dal Server <sub>g</sub>	RF 64
TS 67.1	Viene verificato che l'amministratore possa inserire dei Template <sub>g</sub> di presentazioni	RF 67.1
TS 67.4	Viene verificato che l'amministratore possa inserire Template <sub>g</sub> di Infografiche <sub>g</sub>	RF 67.4
TS 67.7	Viene verificato che l'amministratore possa inserire elementi <sub>g</sub> grafici	RF 67.7
TS 67.10	Viene verificato che l'amministratore possa eliminare un Template <sub>g</sub>	RF 67.10
TS 67.13	Viene verificato che l'amministratore possa ripristinare un Template <sub>g</sub> eliminato	RF 67.13
TS 70.1	Viene verificato che si possa selezionare una presentazione da cui produrre l'Infografica <sub>g</sub>	RF 70.1
TS 70.4	Viene verificato che si possa selezionare Template <sub>g</sub> di Infografica <sub>g</sub>	RF 70.4
TS 70.5	Viene verificato che si possano selezionare gli elementi <sub>g</sub> dell'Infografica <sub>g</sub>	RF 70.5
TS 70.10	Viene verificato che si possa passare in modalità modifica di un'Infografica <sub>g</sub>	RF 70.10
TS 70.10.1	Viene verificato che si possa modificare un Elemento <sub>g</sub> di un'Infografica <sub>g</sub>	RF 70.10.1

Test	Descrizione	Requisito
TS 70.10.1.1	Viene verificato che si possano modificare le dimensioni di un Elemento <sub>g</sub> grafico	RF 70.10.1.1
TS 70.10.1.4	Viene verificato che si possa modificare un Elemento <sub>g</sub> testuale	RF 70.10.1.4
TS 70.10.1.4.1	Viene verificato che si possa modificare il Font <sub>g</sub> di un Elemento <sub>g</sub> testuale	RF 70.10.1.4.1
TS 70.10.1.4.4	Viene verificato che si possa modificare la dimensione del carattere di un Elemento <sub>g</sub> testuale	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
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 di sfondo del testo	RF 70.10.1.4.13
TS 70.10.1.7	Viene verificato che si possa cambiare la posizione di un Elemento <sub>g</sub>	RF 70.10.1.7
TS 70.10.4	Viene verificato che si possa rimuovere lo sfondo dell'Infografica <sub>g</sub>	RF 70.4.4
TS 70.10.7	Viene verificato che si possa inserire uno sfondo nell'Infografica <sub>g</sub>	RF 70.10.7
TS 70.10.10	Viene verificato che si possa inserire un Elemento <sub>g</sub> immagine nell'Infografica <sub>g</sub>	RF 70.10.10
TS 70.10.13	Viene verificato che si possa inserire del testo nell'Infografica <sub>g</sub>	RF 70.10.13
TS 70.10.16	Viene verificato che si possa inserire un Frame <sub>g</sub> nella sua interezza presente nella presentazione	RF 70.10.16

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

Tab 10: Descrizione dei test di sistema per i Requisiti di Qualità e Vincoli

Descrizione	Requisito
Viene verificato che ogni funzionalità dell'applicazione sia documentata	RQ <sub>g</sub> 1, 7
Viene verificato che sia disponibile un tutorial per la creazione delle presentazioni	RQ <sub>g</sub> 4
Viene verificato che sia disponibile la documentazione sui test eseguiti	RQ <sub>g</sub> 10
Viene verificato che il sistema offra la possibilità di eseguire offline le presentazioni	RQ <sub>g</sub> 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 11: 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.

### C.3 Test d'unità

Tab 12: Descrizione dei test di Unità per il  $\text{Server}_g$

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



Test	Descrizione	Metodi	Stato
TU8	viene verificato che il $\text{Server}_g$ 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 $\text{Server}_g$ ritorni un array il cui primo $\text{Elemento}_g$ è corretto rispetto alla presentazione inserita	GET private/api/presentations	success
TU10	viene chiamato il metodo del $\text{Server}_g$ 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 $\text{Server}_g$ 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 $\text{Server}_g$ 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 $\text{Server}_g$ 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 $\text{Server}_g$ 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 <sub>g</sub> in una presentazione in database, viene chiamato il metodo del Server <sub>g</sub> con i parametri corretti e si osserva la cancellazione dell'Elemento <sub>g</sub> dalla presentazione	DELETE private/api/presentation-s/[presname]/delete/[type]/[id]	success
TU16	viene chiamato il metodo del Server <sub>g</sub> per l'inserimento di un Elemento <sub>g</sub> in una presentazione, viene verificato l'inserimento dell'Elemento <sub>g</sub> in database	POST private/api/presentation-s/[presname]/element	success
TU17	viene verificata la modifica dell'Elemento <sub>g</sub> paths di una presentazione in database attraverso la chiamata al Server <sub>g</sub> con dei parametri di esempio	PUT private/api/presentation-s/[presname]/paths	success
TU18	viene inserito un Elemento <sub>g</sub> in una presentazione in database, viene chiamato il metodo del Server <sub>g</sub> per la modifica di questo Elemento <sub>g</sub> 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 <sub>g</sub> verificando che ritorni un'array con il primo Elemento <sub>g</sub> avente il nome dell'immagine inserita	GET private/api/files/image	success
TU20	viene inserito un File <sub>g</sub> immagine, viene chiamato il metodo del Server <sub>g</sub> per la cancellazione di quel File <sub>g</sub> e se ne verifica la cancellazione dalla opportuna cartella del Server <sub>g</sub>	DELETE private/api/files/[image]	success
TU21	viene chiamato il metodo del Server <sub>g</sub> per l'inserimento di un File <sub>g</sub> immagine e si verifica l'inserimento del File <sub>g</sub> nella cartella destinazione sul Server <sub>g</sub>	POST private/api/files/[image]	success

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

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

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

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

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

Tab 14: 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 <sub>g</sub> sia consistente che le presentazioni presenti in database	getPresentationsMeta()	success
TU43	viene chiamata la Funzione <sub>g</sub> 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 <sub>g</sub> 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 <sub>g</sub> per il recupero della presentazione, si verifica il corretto ritorno della Funzione <sub>g</sub>	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 <sub>g</sub> per l'inserimento di un Elemento <sub>g</sub> in una presentazione, viene verificato l'inserimento dell'Elemento <sub>g</sub> in database	newElement( presentation:string , onj:object , callback:function )	success
TU49	viene inserito un Elemento <sub>g</sub> in una presentazione in database, viene chiamato il metodo con i parametri corretti e si osserva la modifica dell'Elemento <sub>g</sub> dalla presentazione	updateElement(presentation :string , onj : object , callback : function)	success
TU50	viene inserito un Elemento <sub>g</sub> in una presentazione in database, viene chiamato il metodo con i parametri corretti e si osserva la cancellazione dell'Elemento <sub>g</sub> 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 15: 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 16: 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

Tab 17: Descrizione dei test di Unità per SlideShowElements

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 <sub>g</sub>	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

Tab 18: Descrizione dei test di Unità per InsertEditRemove

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
TU69	Richiama il InsertEditRemove()::insertFrame e verifica che venga inserito nella presentazione un oggetto Frame <sub>g</sub>	slideShowActions:: InsertEditRemove()::insertFrame(...)	success
TU70	Richiama il InsertEditRemove()::insertImage e verifica che venga inserito nella presentazione un oggetto Image	slideShowActions:: InsertEditRemove()::insertImage(...)	success
TU71	Richiama il InsertEditRemove()::insertSVG e verifica che venga inserito nella presentazione un oggetto SVG	slideShowActions:: InsertEditRemove()::insertSVG(...)	success
TU72	Richiama il InsertEditRemove()::insertAudio e verifica che venga inserito nella presentazione un oggetto Audio	slideShowActions:: InsertEditRemove()::insertAudio(...)	success
TU73	Richiama il InsertEditRemove()::insertVideo e verifica che venga inserito nella presentazione un oggetto Video	slideShowActions:: InsertEditRemove()::insertVideo(...)	success
TU74	Richiama il InsertEditRemove()::editPosition(...) su ogni Elemento <sub>g</sub> della presentazione e verifica che ogni Elemento <sub>g</sub> venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::editPosition(...)	success
TU75	Richiama il InsertEditRemove()::editSize(...) su ogni Elemento <sub>g</sub> della presentazione e verifica che ogni Elemento <sub>g</sub> venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::editSize(...)	success
TU76	Richiama il InsertEditRemove()::editRotation(...) su ogni Elemento <sub>g</sub> della presentazione e verifica che ogni Elemento <sub>g</sub> venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::editRotation(...)	success

Test	Descrizione	Metodi	Stato
TU77	Richiama il InsertEditRemove():editBackground(...) su ogni Elemento <sub>g</sub> della presentazione e verifica che ogni Elemento <sub>g</sub> venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove():editBackground(...)	success
TU78	Richiama il InsertEditRemove():editColor(...) su ogni Elemento <sub>g</sub> della presentazione e verifica che ogni Elemento <sub>g</sub> venga modificato appropriatamente, controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove():editColor(...)	success
TU79	Richiama il InsertEditRemove():editShape(...) su ogni Elemento <sub>g</sub> della presentazione e verifica che ogni Elemento <sub>g</sub> 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

Test	Descrizione	Metodi	Stato
TU85	Richiama il InsertEditRemove()::removeBackground(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeBackground(...)	success
TU86	Richiama il InsertEditRemove()::removeSVG(...), controlla che venga ritornato l'oggetto precedente alla modifica	slideShowActions:: InsertEditRemove()::removeSVG(...)	success