

404NotFound

Premi: better than Prezi.



Piano di Qualifica

Informazioni sul documento

Versione	2.0
Redazione	De Lazzari Enrico Vegro Federico
Verifica	Camborata Marco
Responsabile	Gobbo Ismaele
Uso	Esterno
Ultima modifica	22 Maggio 2015
Lista di distribuzione	404NotFound

Descrizione

Documento riguardante le strategie di verifica adottati dal gruppo 404NotFound e gli obiettivi qualitativi del il progetto Premi.

Registro delle modifiche

Versione	Autore	Data	Descrizione
2.0	Gobbo Ismaele	2015-05-22	Approvazione documento
1.7	Camborata Marco	2015-05-18	Verifica finale del documento
1.6	Vegro Federico	2015-05-14	Correzioni ortografiche e stesura resoconto revisione di progettazione
1.5	De Lazzari Enrico	2015-04-22	Stesura sezione Test di Validazione e tracciamento dei test con i requisiti
1.4	Vegro Federico	2015-04-12	Stesura sezione Test di Sistema e Test di Integrazione
1.3	Camborata Marco	2015-04-02	Verifica delle correzioni richieste dal committente
1.2	Vegro Federico	2015-03-22	Correzione sezione Responsabilità rimandando alla relativa sezione del Piano di Progetto
1.1	De Lazzari Enrico	2015-03-09	Spostamento Sezione Risorse nelle Norme di Progetto
1.0	Vegro Federico	2015-01-22	Approvazione documento
0.7	De Lazzari Enrico	2015-01-22	Verifica finale documento
0.6	Camborata Marco	2015-01-21	Correzione errori ortografici
0.5	De Lazzari Enrico	2015-01-19	Verifica prima stesura documento
0.4	Rettore Andrea	2015-01-15	Stesura gestione amministrativa della revisione e resoconto attività di verifica
0.3	Camborata Marco	2015-01-07	Stesura Versione generale della strategia di verifica e obiettivi di qualità
0.2	Camborata Marco	2015-01-05	Stesura sezione risorse
0.1	Camborata Marco	2015-01-05	Stesura introduzione e scheletro documento

Tabella 1: Storico versioni del documento.

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del Prodotto	5
1.3	Glossario	5
1.4	Riferimenti	5
1.4.1	Normativi	5
1.4.2	Informativi	5
2	Visione Generale della Strategia di Verifica	7
2.1	Organizzazione	7
2.1.1	Analisi dei requisiti	7
2.1.2	Progettazione	8
2.1.3	Realizzazione	8
2.1.4	Validazione	8
2.2	Pianificazione Strategica e Temporale	8
2.3	Responsabilità	9
2.4	Risorse	9
2.5	Strumenti, Tecniche e Metodi	9
2.5.1	Strumenti	9
2.5.2	Tecniche di Analisi	10
2.5.3	Misure e Metriche	12
3	Obiettivi di Qualità	13
3.1	Qualità dei Processi	13
3.2	Qualità del Prodotto	15
3.2.1	Funzionalità	15
3.2.2	Affidabilità	16
3.2.3	Efficienza	16
3.2.4	Usabilità	16
3.2.5	Manutenibilità	17
3.2.6	Portabilità	17
4	Gestione amministrativa della revisione	18
4.1	Comunicazione e risoluzione di anomalie	18
4.2	Tattamento delle discrepanze	18
4.3	Procedure di controllo di qualità di processo	19
A	Pianificazione dei test	20
A.1	Test di sistema	20
A.1.1	Descrizione dei test di sistema	20
A.2	Test di integrazione	23
A.2.1	Descrizione dei test di integrazione	24
A.2.2	Tracciamento componenti – test di integrazione	24
A.3	Test di validazione	25
A.3.1	Test TV1	25
A.3.2	Test TV2	26

A.3.3	Test TV3	26
A.3.4	Test TV4	27
A.3.5	Test TV5	27
A.3.6	Test TV6	29
A.3.7	Test TV7	29
A.3.8	Tracciamento Test di Validazione - Requisiti	30
B	Resoconto dell' Attività di Verifica	32
B.1	Revisione della Documentazione	32
B.2	Tracciamento requisiti	32
B.3	Dettaglio delle verifiche tramite analisi	33
B.4	Revisione dei Requisiti	33
B.5	Revisione di Progettazione	34

Elenco delle tabelle

1	Storico versioni del documento.	1
2	Tabella di tracciamento test di sistema / requisiti	22
3	Tabella test di integrazione	24
4	Tabella tracciamento componente - test di integrazione	25

Elenco delle figure

1	Modello SPY	13
2	Modello PDCA	19
3	Diagramma informale della strategia di integrazione	23

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di illustrare le strategie adottate per implementare i processi di verifica e validazione del lavoro svolto da 404NotFound per assicurare la qualità del progetto Premi e dei processi coinvolti nel suo sviluppo. Per raggiungere gli obiettivi di qualità è necessario un processo di verifica continua sulle attività svolte, per questo motivo il presente documento potrà essere aggiornato in seguito a scelte progettuali del gruppo e/o variazione dei requisiti da parte del Proponente.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di un software di presentazione di slide non basato sul modello di PowerPoint_G, sviluppato in tecnologia HTML5_G e che funzioni sia su desktop che su dispositivo mobile. Il software dovrà permettere la creazione da parte dell'autore e la successiva presentazione del lavoro, fornendo effetti grafici di supporto allo storytelling e alla creazione di mappe mentali.

1.3 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio e ai termini utilizzati nei documenti formali tutti i termini e gli acronimi presenti nel seguente documento che necessitano di definizione saranno seguiti da una “G” in pedice e saranno riportati in un documento esterno denominato Glossario_v1.0.pdf. Tale documento accompagna e completa il presente e consiste in un listato ordinato di termini e acronimi con le rispettive definizioni e spiegazioni.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *NormeDiProgetto_v1.0.pdf*;
- **Capitolato d'appalto C4:** Premi: Software di presentazione “better than Prezi” - <http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf>.

1.4.2 Informativi

- **Piano di Progetto:** *PianoDiProgetto_v1.0.pdf*;
- **Slide dell'insegnamento Ingegneria del Software modulo A:**
<http://www.math.unipd.it/~tullio/IS-1/2014/>;
- **Ingegneria del software - Ian Sommerville - 8a Edizione (2007):**
 - Capitolo 27 - Gestione della qualità;
 - Capitolo 28 - Miglioramento dei processi.
- **Complessità ciclomatica:** http://it.wikipedia.org/wiki/Complessità_ciclomatica;

- **ISO/IEC_G 9126:2001** (inglobato da ISO/IEC_G 25010:2011):
http://www2.cnipa.gov.it/site/_contentfiles/01379900/1379951_ISO_209126.pdf:
 - Systems and software engineering;
 - Systems and software Quality Requirements and Evaluation (SQuaRE);
 - System and software quality models.
- **ISO/IEC_G 15504:1998**: Information Tecnology - Process Assessment, conosciuto come SPICE (Software Process Improvement and Capability Determination): http://www2.cnipa.gov.it/site/_contentfiles/00310300/310320_15504.pdf.

2 Visione Generale della Strategia di Verifica

2.1 Organizzazione

Per garantire la qualità del prodotto in tutte le sue fasi di realizzazione, accertandone la conformità rispetto a quanto emerso durante la fase di Analisi dei Requisiti (vedi allegato *AnalisiDeiRequisiti_v2:0:pdf*), si intende svolgere una costante attività di verifica trasversale a tutte le fasi di sviluppo del progetto.

Per poter effettuare un corretto processo di verifica si è scelto di effettuare le dovute operazioni di controllo ogni volta che il prodotto in esame avrà maturato sostanziali modifiche rispetto alla sua precedente versione. Per quanto riguarda la documentazione questa maturazione si rispecchia nel variare dell'indice di versione dei documenti stessi (vedi documento interno *NormeDiProgetto_v2.0.pdf*, sezione 6.6) e una fase di verifica finale è necessaria affinché un qualsiasi documento possa passare alla fase di approvazione da parte del *Responsabile del Progetto*. E' auspicabile che siano svolte verifiche sui documenti non solo prima dell'approvazione ma anche in fasi intermedie nelle quali il documento può non essere ancora stato completato. Ogni svolgimento di una fase di verifica globale sarà riportata nell'apposito registro delle modifiche. Per assicurare il massimo livello di controllo, tuttavia, un primo controllo sommario sui nuovi contenuti viene svolto dal *Verificatore* ad ogni modifica del documento (per approfondimento vedi documento interno *NormeDiProgetto_v.2.0.pdf* sezione 5.4)

Si è scelto e adottato il metodo "Broken Window Theory" secondo il quale, non appena un errore viene rilevato, questo andrà segnalato e corretto il prima possibile onde evitarne la propagazione.

Il ciclo di vita scelto per lo sviluppo del progetto è un ciclo di vita incrementale (vedi documento allegato *PianoDiProgetto_v2.0.pdf*) e di conseguenza le operazioni di verifica verranno realizzate in modo tale da intervenire in maniera coerente nelle varie fasi del progetto come illustrato di seguito:

2.1.1 Analisi dei requisiti

Tutta la documentazione relativa alla RR, una volta completata, entrerà nella dedicata fase di revisione. Di seguito i parametri di controllo:

- La presenza di eventuali errori lessico/grammaticali e la generale correttezza dei contenuti esposti. Nel dettaglio, il controllo ortografico verrà effettuato con gli strumenti messi a disposizione da TexMaker_G, mentre il controllo lessicale, grammaticale e sintattico da un'accurata rilettura del testo;
- Il controllo dei contenuti con l'obiettivo di verificare la copertura delle richieste del proponente e questo tramite un'accurata rilettura e confronto con il capitolato d'appalto;
- Corrispondenza tra ogni requisito e caso d'uso corrispondente;
- Verifica dei contenuti grafici e tabellari e conformità dei documenti alle *Norme di Progetto* stabilite.

Se durante la verifica saranno state rilevate irregolarità queste verranno segnalate tramite un apposito ticket dal *Verificatore* e corrette dal redattore.

2.1.2 Progettazione

Il processo di verifica in fase di Progettazione consisterà nel verificare che tutti i requisiti descritti durante la fase di Analisi dei Requisiti siano tracciabili nei componenti individuati e viceversa che ogni componente soddisfi o sia associato ad almeno un requisito. Qualora dalla verifica sorgano incongruenze o mancanze, queste verranno segnalate tramite ticket e successivamente risolte.

2.1.3 Realizzazione

La verifica in questa fase verrà effettuata da parte dei programmatori stessi utilizzando appositi e specifici strumenti di verifica automatizzata del codice. La presenza di errori verrà segnalata da un apposito ticket che verrà preso in carico dai programmatori e chiuso una volta risolto il problema.

2.1.4 Validazione

Il team 404NotFound si impegna a garantire il corretto funzionamento del prodotto Premi e a fornire al collaudo una versione funzionante e possibilmente completa del prodotto. Nel caso in cui vengano riscontrati malfunzionamenti o discrepanze tra le caratteristiche del prodotto e le richieste del cliente sarà cura del fornitore eliminare tali difetti, interamente a proprio carico.

2.2 Pianificazione Strategica e Temporale

Avendo l'obiettivo di rispettare le scadenze fissate nel *PianodiProgettov2.0.pdf*, è necessario che l'attività di verifica della documentazione e del codice sia sistematica e ben organizzata. Ogni fase di redazione dei documenti e di codifica deve essere preceduta da una fase di studio preliminare per eliminare all'origine possibili imprecisioni di natura concettuale e/o tecnica.

Il processo di verifica viene strutturato in tre fasi:

1. **Pre-Verifica:** Si tratta della pianificazione e la preparazione delle attività di verifica. Consiste nella scelta delle persone che si occuperanno di questa attività e nella distribuzione dei documenti o componenti software da controllare;
2. **Verifica effettiva:** I *Verificatori* lavorano indipendentemente per trovare errori, omissioni e scostamenti rispetto agli standard, durante questa fase, un autore del documento o componente software attende il responso del *Verificatore*. Deve stillato un elenco delle azioni correttive da intraprendere;
3. **Post-Verifica:** Dopo che le correzioni sono state apportate al componente in esame il *Verificatore* usando come checklist l'elenco delle correzioni da lui redatto nella fase precedente, potrà constatare l'avvenuta correzione.

Durante le attività di verifica è inevitabile che gli errori commessi dagli individui vengano esposti a tutto il gruppo. E' quindi molto importante che si incoraggi nel team una mentalità per la quale la segnalazione degli errori non diventi motivo per screditare il lavoro di un singolo, ma occasione di crescita per la persona e per l'intero gruppo di lavoro.

2.3 Responsabilità

Per garantire che il processo di verifica sia efficace e sistematico vengono attribuite delle responsabilità ad ogni specifico ruolo del progetto. I ruoli che detengono le responsabilità del processo di verifica sono il *Responsabile del Progetto* ed i *Verificatori*. La suddivisione dei compiti e le modalità di attuazione degli stessi sono definiti nel documento *NormeDiProgetto_v2.0.pdf*.

2.4 Risorse

Per assicurare che gli obiettivi qualitativi vengano raggiunti e` necessario l'utilizzo di risorse sia umane che tecnologiche. Coloro che detengono la responsabilità maggiore per l'attività di verifica e validazione sono il *Responsabile del Progetto* e il *Verificatore*. Per una dettagliata descrizione dei ruoli e delle loro responsabilità fare riferimento alle *Norme di Progetto*. Per risorse tecniche e tecnologiche sono da intendersi tutti gli strumenti software e hardware che il gruppo intende utilizzare per attuare le attività di verifica su processi e prodotti.

2.5 Strumenti, Tecniche e Metodi

2.5.1 Strumenti

Per lo svolgimento del processo di verifica faremo uso dei seguenti strumenti:

- **Correttore automatico di TeXMaker_G**: come segnalato nelle *NormeDiProgetto_v2.0.pdf* per la scrittura di documenti si è scelto di utilizzare l'ambiente grafico TeXMaker_G. Tale strumento integra i dizionari di OpenOffice.org e segnala i potenziali errori ortografici presenti nel testo;
- **404TrackerDB**: Strumento software realizzato dal gruppo 404NotFound che contiene ed associa:
 - Requisiti individuati durante l'analisi;
 - Fonti di requisiti individuate, inclusi anche i casi d'uso.

Permette inoltre di esportare automaticamente:

- Codice \LaTeX per la descrizione dei casi d'uso;
 - Tabella in \LaTeX per il tracciamento fonti-requisiti.
- Strumenti W3C_G (www.w3.org) per la validazione:
 - **validatore HTML5_G** (<http://validator.w3.org>);

- **validatore CSS_G** (<http://jigsaw.w3.org/css-validator/>).
- Strumenti per debugging_G HTML_G, CSS_G e JavaScript_G messi a disposizione dai vari browser_G:
 - **Chrome Developer Tools** (<https://developers.google.com/chrome-developer-tools>);
 - **Firebug** (<http://getfirebug.com/>).
- ;
- **JSLint** Ambiente di test (<http://www.jshint.org>): tool per la validazione di codice JavaScript_G;
- **JUnit** (<http://www.junit.org>): semplice framework per eseguire test ripetibili;
- **BrowserStack** (<http://www.browserstack.com/>): per eseguire il test comparato sui vari browser_G;
- **WebStorm** (<https://www.jetbrains.com/webstorm/>): IDE JavaScript_G scelto come ambiente di sviluppo.

2.5.2 Tecniche di Analisi

Anamalisi Statica:

Consiste nell'analisi della documentazione e dei prodotti software senza effettuare l'esecuzione. Viene svolta mediante due tecniche complementari:

- **Walkthrough:** È una tecnica che viene utilizzata soprattutto nelle prime fasi del progetto, quando ancora non è stata maturata una adeguata esperienza da parte dei membri del gruppo, che permetta di attuare una verifica più mirata e precisa. Consiste nella rilettura completa e metodica da parte dell'autore stesso o da parte del *Verificatore* allo scopo di trovare errori. 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. 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;
- **Inspection:** Questa tecnica consiste nell'analisi di alcune parti del documento o del codice alla ricerca di errori solo in parti ritenute critiche in base all'esperienza derivata dalle revisioni precedenti. La lista di controllo o checklist, che deve essere seguita per svolgere efficacemente questo processo, deve essere redatta anticipatamente ed è frutto del lavoro svolto dai verificatori con la tecnica di walkthrough. L'Inspection è da preferire al Walkthrough, poiché non necessita della lettura integrale dei documenti in oggetto, ma richiede un sufficiente livello di dettaglio nella lista di controllo.

Metodi di Analisi Statica:

- **Analisi del flusso di controllo:** si controlla che il codice sia correttamente strutturato e che segua il flusso aspettato, che non vi siano parti del programma che possano non terminare e che non esistano porzioni di codice non raggiungibile;
- **Analisi del flusso dei dati:** si accerta che il software non acceda mai a variabili non inizializzate o non modifichi più volte di seguito una variabile senza leggerne il valore tra una modifica e l'altra;
- **Analisi del flusso di informazione:** si verifica che le uniche dipendenze tra gli input e gli output di ogni unità di codice o di più unità siano quelle previste in fase di progettazione.

Analisi Dinamica:

Consiste nella verifica dei componenti del software o del sistema in generale e richiede l'esecuzione del programma per eseguire il test. Perché tale attività sia utile e generi risultati attendibili è necessario che i test effettuati siano ripetibili: cioè dati uno stesso input e uno stesso ambiente di esecuzione deve fornire gli stessi risultati quando vengono effettuate più prove. Questi risultati saranno utili solo se porteranno alla luce errori permettendo di correggerli, nel caso non vengano riscontrate anomalie, ciò non costituisce una prova dell'assenza di errori.

Metodi di Analisi Dinamica:

- **Test di unità:** Viene verificata ogni unità software che deve soddisfare i requisiti per essa richiesti ed è necessario testare tutte le possibili esecuzioni del codice che la compongono. Per unità si intende la più piccola quantità di software che è utile verificare singolarmente e che viene prodotta da un singolo *Programmatore*;
- **Test di integrazione:** I moduli che hanno superato il test di unità possono venire integrati tra di loro. Il test di integrazione ha lo scopo di individuare errori residui nella realizzazione dei moduli e problemi nell'integrazione con altre componenti fornite da terze parti che non si conoscono a fondo;
- **Test di sistema e collaudo:** Serve a verificare il completo soddisfacimento dei requisiti software stabiliti in fase di Analisi. Consiste nella validazione del prodotto software nel momento in cui vengono aggiunti tutti i componenti. Il test potrà riguardare, in una fase iniziale, solamente alcune delle componenti del prodotto finale, per poi interessare il sistema nella sua interezza;
- **Test di regressione:** Consiste nell'eseguire nuovamente i test di unità e di integrazione su componenti software alle quali sono stati apportati cambiamenti. Serve a controllare che le modifiche apportate non provochino malfunzionamenti alla componente stessa o ad altre che dipendono da essa;
- **Test di accettazione:** È il test di collaudo del prodotto software che viene eseguito in presenza del Committente. Se questa fase finale di test viene superata positivamente si può procedere al rilascio ufficiale del prodotto sviluppato.

2.5.3 Misure e Metriche

Le misure e le metriche che il team adotterà si ispireranno alle indicazioni dello standard ISO/IEC_G-14598. Tale norma descrive il processo di valutazione della qualità del software. Vengono di seguito descritte le metriche sulle quali 404NotFound intende basarsi nei processi di verifica, sia in fase di progettazione che di codifica, che potranno essere integrate e stabilite con maggiore precisione durante l'avanzamento del progetto.

- **Complessità ciclomatica:** La complessità ciclomatica di un programma è il numero di cammini linearmente indipendenti attraverso il codice sorgente. Per esempio, se il codice sorgente non contiene IF o FOR, allora il livello di complessità sarà 1, poiché esiste un solo cammino, se è presente un IF la complessità diventa di 2. Un programma si può quindi rappresentare come un albero nel quale i nodi sono i blocchi del programma e gli archi sono il passaggio del controllo da un blocco all'altro. La complessità è quindi definita come:

$$C = e - n + 2p$$

dove e è il numero di archi, n il numero di nodi e p il numero delle componenti connesse. Il valore massimo di complessità ciclomatica accettabile per il gruppo di lavoro 404NotFound è 10;

- **Misure nella progettazione:**

- **Complessità di flusso:** misura la quantità di informazioni in entrata ed uscita da una funzione (fan in e fan out). Fan-in è una misura del numero di metodi che invocano una determinata procedura. Un alto valore per fan-in significa che cambiamenti a quella procedura potrebbero avere effetti a catena sulle altre. Fan-out indica quanto una procedura ne richiama delle altre, dando una valutazione del grado di dipendenza di quella procedura dalle altre;
- **Profondità di annidamento dei costrutti condizionali:** costrutti IF profondamente annidati sono più soggetti a errori e risultano più difficili da comprendere e da correggere.

- **Misure sul codice:**

- **Lunghezza del codice:** misura la dimensione di un componente in termini di numero di linee di codice. Generalmente, maggiore è la dimensione di un componente, maggiore è la probabilità che esso contenga degli errori;
- **Lunghezza degli identificatori:** misura la lunghezza media degli identificatori (variabili, classi, metodi, etc.). Una lunghezza media elevata è indice di un elevato grado di complessità;
- **Numero dei parametri:** indica il numero di parametri formali nei metodi. Quando questo indice è troppo elevato, è opportuno pensare di semplificare i metodi suddividendoli in metodi più semplici. Il numero massimo di parametri per metodo stabilito dal team di sviluppo è 6.

3 Obiettivi di Qualità

Il gruppo 404NotFound ha ritenuto importante fissare alcuni obbiettivi di qualità da perseguire nel prodotto finale e nei processi di realizzazione, questo per garantire una migliore e più efficace soddisfazione dei requisiti richiesti nel capitolato d'appalto.

3.1 Qualità dei Processi

Per garantire la qualità del prodotto è necessario perseguire la qualità dei processi che lo definiscono. Per fare questo il team 404NotFound ha deciso di adottare lo standard ISO/IEC_G 15504 denominato SPICE_G (Software Process Improvement Capability Determination), il quale definisce il modello denominato SPY_G (SW Process Assessment & Improvement, vedi Figura 1), per la valutazione dei processi in un'organizzazione del settore IT (Information Technology).

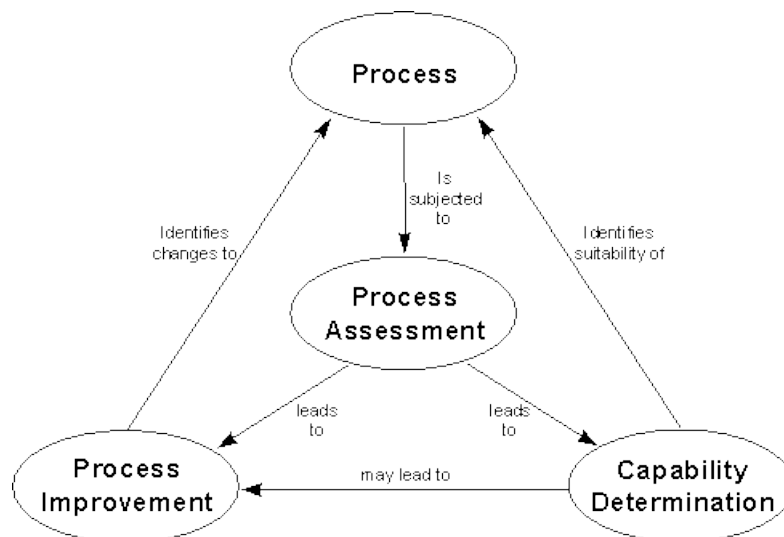


Figura 1: Modello SPY

Lo standard identifica e definisce nove attributi di qualità:

1. **Process performance attribute:** un processo raggiunge i suoi obiettivi, trasformando input identificabili in output identificabili;
2. **Performance management attribute:** l'attuazione di un processo è pianificata e controllata al fine di produrre risultati che rispondano agli obiettivi attesi;
3. **Work product management attribute:** capacità del processo di elaborare un prodotto documentato, controllato e verificato;
4. **Process definition attribute:** l'esecuzione del processo si basa su standard di processo per raggiungere i propri obiettivi;
5. **Process resource attribute:** capacità del processo di attingere a risorse tecniche e umane appropriate per essere attuato efficacemente;

6. **Process measurement attribute:** i risultati raggiunti e le misure rilevate durante l'attuazione di un processo sono stati usati per assicurarsi che l'attuazione di tale processo supporti efficacemente il raggiungimento di specifici obiettivi;
7. **Process control attribute:** il processo viene controllato attraverso la raccolta, analisi ed utilizzo delle misure di prodotto e di processo, al fine di correggere, se necessario, le sue modalità di attuazione;
8. **Process change attribute:** i cambiamenti strutturali, di gestione e di esecuzione vengono gestiti in modo controllato;
9. **Continuous improvement attribute:** le modifiche al processo sono identificate e implementate per garantire il miglioramento continuo nella realizzazione degli obiettivi di business dell'organizzazione.

La norma definisce poi quattro livelli di possesso di ciascun attributo:

- **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): vi è evidenza di approccio sistematico al raggiungimento del possesso di un attributo, ma alcuni aspetti del possesso possono essere non prevedibili;
- **L** - Largamente posseduto (51%-85% di possesso): vi è evidenza di approccio sistematico al raggiungimento di un significativo livello di possesso di un attributo, ma l'attuazione del processo può variare nelle diverse unità operative della organizzazione;
- **F** - (Fully) Pienamente posseduto (86%-100% di possesso): vi è evidenza di un approccio completo e sistematico e di un pieno raggiungimento del possesso dell'attributo, non esistono significative differenze nel modo di attuare il processo tra le diverse unità operative.

Vi sono infine cinque livelli di maturità di processi:

- **Livello 0 - Processo incompleto:** il processo non è implementato o non raggiunge gli obiettivi. Non vi è evidenza di approcci sistematici agli attributi definiti;
- **Livello 1 - Processo semplicemente attuato:** il processo viene messo in atto e raggiunge i suoi obiettivi. Non vi è evidenza di un approccio sistematico ad alcuno degli attributi definiti. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di "process performance";
- **Livello 2 - Processo gestito:** il processo 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 definito:** il processo è attuato, pianificato e controllato sulla base di procedure ben definite, basate sui principi del software engineering. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process definition” e “Process resource”;
- **Livello 4 - Processo predicibile:** il processo è stabilizzato ed è attuato all’interno di definiti limiti riguardo i risultati attesi, le performance, le risorse impiegate ecc. Il raggiungimento di questo livello è dimostrato attraverso il possesso degli attributi di “Process measurement” e “Process control”;
- **Livello 5 - Processo ottimizzante:** il processo 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”.

Adottando lo standard ISO/IEC_G 15504 gli sviluppatori software del gruppo 404NotFound possono e intendono ottimizzare l’uso delle risorse e contenere così i costi con una migliore stima dei rischi e la possibilità di confrontarsi con delle best practice_G.

3.2 Qualità del Prodotto

Per quanto concerne la qualità del prodotto si è scelto di seguire alcune linee guida dettate dallo standard ISO/IEC_G 9126, che definisce la qualità del prodotto software come l’insieme delle caratteristiche che incidono sulla capacità del prodotto di soddisfare requisiti espliciti o impliciti. Tale standard individua sei caratteristiche indicatrici di qualità del prodotto software, ciascuna delle quali suddivisa in sotto-caratteristiche.

3.2.1 Funzionalità

Il prodotto software realizzato deve offrire apposite funzionalità che siano in grado di soddisfare requisiti funzionali espliciti o impliciti. Le sue sotto-categorie sono:

- **Appropriatezza:** capacità di offrire un insieme di funzioni appropriate per i compiti e gli obiettivi prefissati all’utente;
- **Accuratezza:** capacità del software di fornire i risultati concordati o i precisi effetti richiesti;
- **Interoperabilità:** capacità di interagire ed operare con altri sistemi;
- **Conformità:** capacità di aderire a standard, convenzioni e regolamentazioni rilevanti al settore operativo in cui viene applicato;
- **Sicurezza:** capacità di proteggere informazioni e dati impedendo gli accessi e le modifiche non autorizzati, mentre garantendo queste operazioni a utenti o sistemi autorizzati.

Per misurare il raggiungimento di questo obiettivo si verificherà la quantità di requisiti soddisfatti che avranno un riscontro in elementi funzionanti nell’applicazione prodotta. La soglia di sufficienza è il soddisfacimento di tutti i requisiti obbligatori previsti dal capitolato d’appalto.

3.2.2 Affidabilità

L'affidabilità misura la capacità di un prodotto software di mantenere un determinato livello di prestazioni se usato in determinate condizioni e per un certo periodo.

- **Maturità:** capacità di evitare che si verifichino fallimenti o malfunzionamenti a causa di errori nel software;
- **Tolleranza agli errori:** capacità di mantenere determinati livelli di prestazioni nonostante l'insorgere di errori, malfunzionamenti o un uso scorretto del prodotto;
- **Recuperabilità:** capacità di ripristinare il livello appropriato di performance e di recuperare le informazioni o dati rilevanti in seguito all'insorgere di un'anomalia;
- **Aderenza:** capacità di aderire a standard, convenzioni e regolamentazioni inerenti l'affidabilità.

Per misurare il raggiungimento di questo obiettivo si calcolerà il numero di esecuzioni totale confrontandolo con quelle andate a buon fine e che hanno mantenuto un livello di prestazioni tali da poter permettere l'utilizzo previsto del prodotto.

3.2.3 Efficienza

L'efficienza si misura mettendo in relazione la capacità di fornire prestazioni adeguate con la quantità di risorse impiegate.

- **Comportamento rispetto al tempo:** capacità di fornire tempi di risposta e di elaborazione adeguati per le funzioni richieste, sotto condizioni determinate;
- **Utilizzo delle risorse:** capacità di utilizzare in maniera adeguata la giusta quantità e tipologia di risorse.

Il raggiungimento di questo obiettivo sarà misurato dal tempo necessario per ottenere una risposta dal servizio (risposta dell'applicazione più il tempo necessario alla connessione) in condizioni normali e in condizioni di sovraccarico.

3.2.4 Usabilità

L'usabilità di un prodotto software si determina in base alla sua capacità di essere capito, appreso e usato dall'utente.

- **Comprensibilità:** costituisce la facilità di comprensione dei concetti del prodotto, permettendo all'utente quindi di comprendere se il programma è appropriato e come può essere utilizzato per compiti specifici;
- **Apprendibilità:** capacità di diminuire l'impegno richiesto agli utenti per imparare ad utilizzare l'applicazione;
- **Operabilità:** capacità di porre gli utenti in condizioni tali da utilizzare il prodotto e controllarne l'uso;

- **Attrattività:** capacità di essere piacevole e di creare interesse nell'utente;
- **Conformità:** capacità di adesione a standard o convenzioni relativi all'usabilità.

Il raggiungimento di questo obiettivo sarà misurato in base alla capacità dell'applicativo di adattarsi ai vari tipi di ambienti in cui esso verrà eseguito (ambienti desktop o dispositivi mobile). L'usabilità sarà poi ritenuta raggiunta fornendo un'interfaccia il più possibile chiara, semplice ed intuitiva per l'utente.

3.2.5 Manutenibilità

La manutenibilità rappresenta la capacità del software di subire modifiche di natura correttiva, miglioramenti o adattamenti, con un impegno contenuto.

- **Analizzabilità:** capacità di facilitare l'analisi del codice e limitare l'impegno richiesto per localizzare un eventuale errore;
- **Modificabilità:** capacità del prodotto di permettere l'implementazione di una specificata modifica;
- **Stabilità:** capacità di evitare effetti inaspettati a seguito delle modifiche apportate;
- **Testabilità:** capacità di essere facilmente testato per validare le modifiche apportate.

La misurazione del raggiungimento di questo obiettivo sarà legata al rispetto delle misure e metriche descritte nel capitolo 2.4.3.

3.2.6 Portabilità

La portabilità è la capacità di un software d'essere trasferito da un ambiente di lavoro ad un altro.

- **Adattabilità:** capacità di essere adattato per differenti ambienti operativi eliminando o limitando la necessità di applicare modifiche;
- **Installabilità:** capacità di richiedere il minor impegno possibile per essere installato in uno specifico ambiente;
- **Coesistenza:** capacità di coesistere condividendo risorse con altri software nel medesimo ambiente;
- **Sostituibilità:** capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti nello stesso ambiente.

L'obiettivo dovrà essere raggiunto ottenendo la compatibilità con i principali browser_G (Google Chrome, FireFox e Internet Explorer) e validando il codice secondo gli standard del W3C_G.

4 Gestione amministrativa della revisione

4.1 Comunicazione e risoluzione di anomalie

Un'anomalia consiste in un comportamento non coerente con le aspettative. Un esempio di anomalie che possono essere riscontrate sono:

- Violazione delle norme tipografiche da parte di un documento;
- Incongruenza del prodotto con funzionalità indicate nell'analisi dei requisiti.

Lo strumento scelto per la gestione delle anomalie è la sezione “Issue” messa a disposizione da Github_G. Coerentemente con l'organizzazione generale delle strategie di verifica, nuove anomalie potranno essere scoperte in due modi:

- In ogni fase di verifica, il *Verificatore* avrà il compito di cercare eventuali anomalie;
- Grazie all'approccio “Broken Window Theory” (vedi sezione 2.1), chiunque in qualunque momento è incentivato alla ricerca di possibili anomalie.

Nel caso in cui un *Verificatore* o un membro del gruppo individui un'anomalia dovrà segnalarlo aprendo un ticket_G (vedi documento allegato *NormeDiProgetto_v1.0.pdf* sezione 5.2). Un *Verificatore* ha il compito di controllare le pull request quindi nel caso trovasse un'anomalia deve impedire la pull request con le modalità descritte nella sezione 5.4 delle *NormeDiProgetto_v1.0.pdf*.

4.2 Trattamento delle discrepanze

Una discrepanza è un discostamento dai requisiti attesi del capitolato o una violazione delle Norme di Progetto. Il trattamento delle discrepanze avviene come la gestione delle anomalie. Quando un membro del gruppo o il *Verificatore* ne individuasse una segnalerà il problema aprendo un ticket_G oppure un *Verificatore* può bloccare la pull specificando il motivo al richiedente come per il trattamento delle anomalie.

4.3 Procedure di controllo di qualità di processo

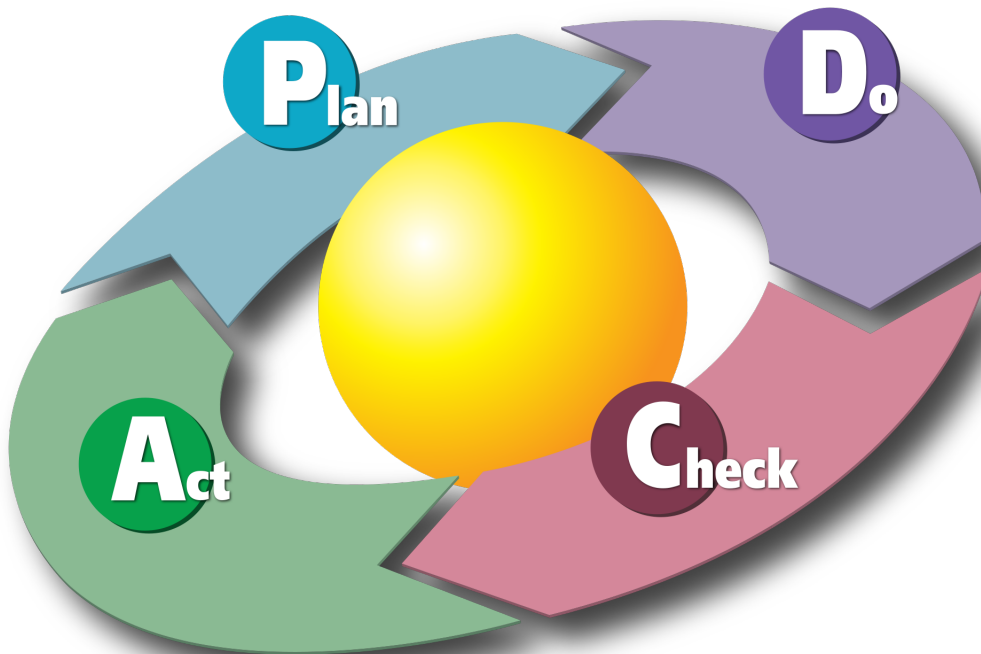


Figura 2: Modello PDCA

Le Procedure di controllo di qualità di processo si basano sul ciclo di Deming o $PDCA_G$. Questo garantisce un miglioramento continuo di tutti i processi e delle attività di verifica che si realizza con comunicazioni attive delle componenti del gruppo e con la connessione delle fasi di analisi, progettazione, verifica e collaudo. La qualità dei processi viene monitorata anche grazie alla qualità di prodotto perché un prodotto di bassa qualità può indicare che uno o più processi vadano migliorati. Per questo motivo si presta attenzione a monitorare i singoli processi ed è necessario quindi che i processi vengano pianificati nel dettaglio, le risorse vengano ripartite nella pianificazione in modo chiaro e ci sia un adeguato controllo sui processi.

A Pianificazione dei test

Si descrivono di seguito tutti i test di validazione, sistema ed integrazione previsti, prevedendo un aggiornamento futuro per i test di unita`. Per le tempistiche di esecuzione dei test si faccia riferimento al *Piano di Progetto* v2.0. Nelle tabelle sottostanti lo stato dei test "Pianificato" e` da intendersi come non applicato in quanto tali test saranno applicati successivamente, come descritto nel *Piano di Progetto*.

A.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 descritti nell'*Analisi dei Requisiti* v2.0. I test di sistema riportati sono quelli relativi ai requisiti software individuati e pertanto meritevoli di un test.

A.1.1 Descrizione dei test di sistema

Requisito	Stato	Descrizione	Test
FOb1	Pianificato	Si verifica che il sistema permetta la creazione di una presentazione e che questa venga effettivamente creata e salvata sul database	TS1
FOb3	Pianificato	Si testa l'esecuzione di una presentazione, verificando che il percorso seguito sia quello scelto dall'utente ed il corretto avanzamento delle slides	TS3
FOb3.5	Pianificato	Si verifica il funzionamento dei checkpoint _G , la possibilità per l'utente di impostare un checkpoint _G e il corretto funzionamento dei percorsi di approfondimento	TS3.5
FOb4	Pianificato	Si verifica che le modifiche apportate dall'utente ad una presentazione, quali ad esempio inserimento di oggetti grafici, testi o nuovi frame _G vengano correttamente applicate e salvate dal sistema	TS4
FOb5	Pianificato	Viene verificato che il salvataggio della presentazione avvenga con successo e nel formato atteso	TS5

FOb6	Pianificato	Si verifica che l'azione di eliminazione da parte di un'utente di una presentazione cancelli effettivamente tutti i dati relativi a quella presentazione dal sistema	TS6
FOb6.1	Pianificato	Si testa l'annullamento del comando di eliminazione e si verifica che non vi sia alcuna modifica alla presentazione fintanto che l'azione non viene confermata al sistema dall'utente.	TS6.1
FOb7	Pianificato	Si verifica che il sistema permetta la pubblicazione di una presentazione e che questa vada a buon fine	TS7
FOb8	Pianificato	Il sistema deve poter generare un link per una presentazione live	TS8
FOp9	Pianificato	l'utente deve poter rendere privata una presentazione pubblica	TS9
FOb11.1	Pianificato	Viene verificato che il sistema permetta di esportare la presentazione in formato poster controllando che il file di output corrisponda alla presentazione esportata	TS11.1
FOb11.2	Pianificato	Si verifica che il formato con cui viene esportata la presentazione sia portabile	TS11.2
FOb13	Pianificato	Si verifica che il sistema permetta all'utente di registrarsi simulando i passi della procedura di registrazione e successivamente verificando il salvataggio dei dati inseriti dall'utente	TS13
FOb14	Pianificato	Si verifica che il sistema permetta all'utente di autenticarsi	TS14

FOb15	Pianificato	Viene verificato il corretto funzionamento di tutti i messaggi di errore da parte del sistema, l'utente deve sapere quando ha commesso un errore	TS15
FOb16	Pianificato	Viene verificata e provata la procedura di cambio password, assicurandosi che il sistema salvi correttamente le modifiche dell'utente	TS16
VOb17	Pianificato	Viene testato l'avvio del sistema su browser _G Chrome da versione 40+	TS17

Tabella 2: Tabella di tracciamento test di sistema / requisiti

A.2 Test di integrazione

In questa sezione vengono descritti i test di integrazione per i vari componenti descritti nella progettazione ad alto livello, che permettono di verificare la corretta integrazione ed il corretto flusso dei dati all'interno del sistema. Si è scelta una strategia di integrazione incrementale, il cui principale vantaggio è quello di poter sviluppare e verificare le componenti in parallelo. Con l'approccio incrementale, infatti, i difetti rilevati da un test sono da attribuirsi, con maggior probabilità, all'ultima componente aggiunta ed essendo ogni passo di integrazione reversibile è possibile retrocedere ed effettuare un rollback ad uno stato noto e sicuro. Per i test di integrazione è stato utilizzato il metodo bottom-up, questa scelta risulta automatica avendola già adottata nella progettazione delle componenti con l'obiettivo quello di ridurre le dipendenze funzionali di ogni singola componente. Il diagramma seguente non rispetta il formalismo UMLG 2.x ed è utilizzato per semplificare l'illustrazione della strategia di integrazione. Si possono notare, lungo l'albero, dei macro componenti che possono integrare al loro interno due o più componenti di maggior dettaglio.

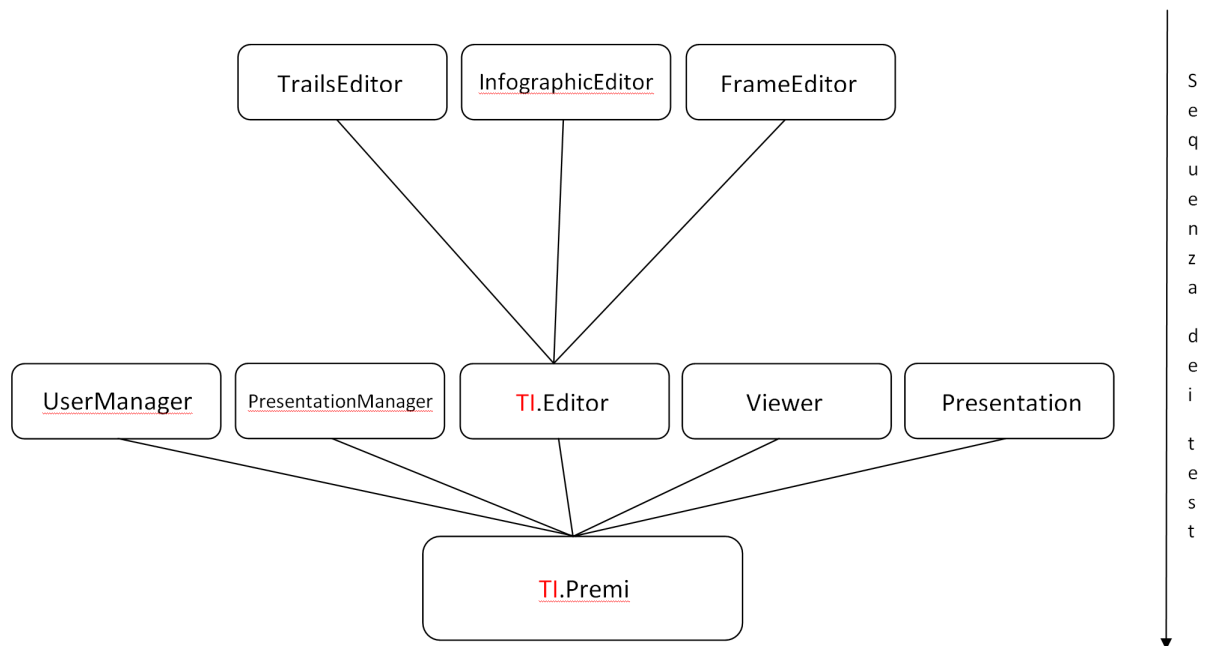


Figura 3: Diagramma informale della strategia di integrazione

A.2.1 Descrizione dei test di integrazione

Test	Descrizione	Stato
TI.Premi	Test di integrazione finale per le componenti del modulo Premi	Pianificato
TI.UserManager	Gestione dell'Utente: Verifica il corretto funzionamento delle operazioni di registrazione, autenticazione e cambio password	Pianificato
TI.PresentationManager	Gestione della presentazione: Verifica il corretto funzionamento del sistema di gestione della presentazione, testa la correttezza delle operazioni di creazione, modifica eliminazione, pubblicazione, esportazione e portabilità di una presentazione	Pianificato
TI.Editor	Test di integrazione finale per le componenti del modulo Premi.Editor	Pianificato
TI.TrailsEditor	Edit dei percorsi: Verifica il corretto funzionamento dei percorsi, e le procedure di creazione, modifica, clonazione ed eliminazione di un percorso	Pianificato
TI.InfographicEditor	Edit dell'Infografica: Verifica il corretto funzionamento delle infografiche, nello specifico l'aggiunta di $frame_G$, immagini, testi e $shape_G$ all'infografica e la modifica dello stile	Pianificato
TI.FrameEditor	Edit dei Frame: Verifica il corretto funzionamento dei $frame_G$, nello specifico l'aggiunta di immagini, testi e $shape_G$ al $frame_G$ e la modifica dello stile	Pianificato
TI.Viewer	Visualizzazione della presentazione: Verifica la corretta riproduzione della presentazione	Pianificato

Tabella 3: Tabella test di integrazione

A.2.2 Tracciamento componenti – test di integrazione

Nella tabella seguente è riportato il tracciamento delle singole componenti del sistema con il relativo test di integrazione. Si noti che l'integrazione di Views e Controllers sono garantite dall'architettura del sistema.

Componente	Test
Premi	TI.Premi
Premi.UserManager	TI.UserManager
Premi.UserManager.Views	Architettura del Sistema

Premi.UserManager.Controllers	Architettura del Sistema
Premi.PresentationManager	TI.PresentationManager
Premi.PresentationManager.Views	Architettura del Sistema
Premi.PresentationManager.Controllers	Architettura del Sistema
Premi.Editor	TI.Editor
Premi.Editor.Views	Architettura del Sistema
Premi.Editor.Controllers	Architettura del Sistema
Premi.Editor.TrailsEditor	TI.TrailsEditor
Premi.Editor.TrailsEditor.Views	Architettura del Sistema
Premi.Editor.TrailsEditor.Controllers	Architettura del Sistema
Premi.Editor.InfographicEditor	TI.InfographicEditor
Premi.Editor.InfographicEditor.Views	Architettura del Sistema
Premi.Editor.InfographicEditor.Controllers	Architettura del Sistema
Premi.Editor.FrameEditor	TI.FrameEditor
Premi.Editor.FrameEditor.Views	Architettura del Sistema
Premi.Editor.FrameEditor.Controllers	Architettura del Sistema
Premi.Viewer	TI.Viewer
Premi.Viewer.Views	Architettura del Sistema
Premi.Viewer.Controllers	Architettura del Sistema

Tabella 4: Tabella tracciamento componente - test di integrazione

A.3 Test di validazione

In questa sezione vengono descritti i test di validazione che servono per accertarsi che il prodotto realizzato sia conforme alle attese. Per ogni test vengono descritti i vari passi che un utente deve eseguire per testare i requisiti ad esso associati.

A.3.1 Test TV1

L'utente vuole verificare che ci si possa registrare al sistema Premi.

All'utente è richiesto di:

- Aprire il sistema
- Navigare nell'area di registrazione
- Inserire un indirizzo email.
All'utente è chiesto di:
 - Verificare che l'inserimento di un indirizzo email non valido generi un avviso da parte del sistema
 - Inserire un indirizzo email valido
- Inserire una password
- Reinserire la password per conferma
- Verificare che il completamento della registrazione al sistema vada a buon fine

A.3.2 Test TV2

L'utente vuole verificare il corretto funzionamento dell'autenticazione al sistema Premi
All'utente è richiesto di:

- Eseguire la registrazione al sistema Premi eseguendo **TV1**.
- Accedere alla sezione di Login del sistema Premi
- Inserire un indirizzo email
All'utente è chiesto di:
 - Verificare che l'inserimento di un indirizzo email non valido generi un avviso da parte del sistema
 - Provare l'inserimento di un indirizzo email diverso da quello fornito in fase di registrazione
- Inserire una password
All'utente è chiesto di:
 - Provare l'inserimento di una password diversa da quella fornita in fase di registrazione
 - Testare la procedura guidata di cambio password (**TV2.1**)
- Verificare che in caso di indirizzo mail o password diversi da quelli forniti in fase di registrazione generi un messaggio di errore non facendo terminare a buon fine l'autenticazione.
- Verificare che l'autenticazione vada a buon fine e il sistema renda esplicito il successo di questa azione.

A.3.3 Test TV3

L'utente vuole testare la creazione di una nuova presentazione e successivamente la sua eliminazione

All'utente è richiesto di:

- Autenticarsi al sistema Premi eseguendo **TV2**
- Creare una nuova presentazione
- Scegliere ed inserire un titolo per la nuova presentazione (**TV.3.1**)
- Scrivere una descrizione di almeno di almeno 15 parole per la presentazione (**TV.3.2**)
- Completare la procedura di creazione confermando i dati inseriti
- Constatare l'avvenuta creazione della presentazione
- Selezionare la presentazione
- Eliminare la presentazione selezionata
- Annullare l'operazione di eliminazione prima che questa avvenga effettivamente (**TV.3.3**)

A.3.4 Test TV4

L'utente vuole testare l'esecuzione di una presentazione

All'utente è richiesto di:

- Autenticarsi al sistema Premi eseguendo **TV2**
- Selezionare una presentazione dall'elenco delle presentazioni (**TV4.1**)
- Scegliere un percorso per la presentazione tra quelli disponibili (**TV4.2**)
- Navigare nella presentazione:
All'Utente è chiesto di:
 - Avanzare nel percorso presentativo un passo alla volta (**TV4.3**)
 - Retrocedere nel percorso presentativo un passo alla volta (**TV4.4**)
 - Seguire un percorso di approfondimento a partire da un checkpoint_G (**TV4.5**)
 - Tornare ad un checkpoint_G dopo aver completato un percorso di approfondimento (**TV.4.6**)
- Interrompere l'esecuzione della presentazione (**TV4.7**)

A.3.5 Test TV5

L'utente vuole testare la modifica di una presentazione e il salvataggio delle modifiche

All'utente è richiesto di:

- Autenticarsi al sistema Premi eseguendo **TV2**
- Selezionare una presentazione dall'elenco delle presentazioni (**TV4.1**)
- Entrare nell'editor
- Inserire nuovi oggetti grafici nella presentazione (**TV5.1**)
All'utente è chiesto di:
 - Provare l'inserimento di un'area di testo (**TV5.2**)
All'utente è chiesto di:
 - * Inserire un testo di almeno 10 parole (**TV5.2.1**)
 - * Scegliere un font per il testo (**TV5.2.2**)
 - * Scegliere un colore per il testo tra quelli disponibili (**TV5.2.3**)
 - * Scegliere una dimensione diversa da quella di default per il testo (**TV5.2.4**)
 - Provare l'inserimento di un frame_G (**TV5.3**) Fob4.1.2
All'utente è chiesto di:
 - * Scegliere una forma tra quelle disponibili per il frame_G (**TV5.3.1**)
 - Provare l'inserimento di un'immagine (**TV5.4**)
All'utente è chiesto di:
 - * Scegliere un file immagine da filesystem (**TV5.4.1**)

- Provare l’inserimento di uno shape_G (**TV5.5**)
All’utente è chiesto di:
 - * Scegliere una forma tra quelle disponibili per lo shape_G (**TV5.5.1**)
- Selezionare un oggetto grafico tra quelli precedentemente inseriti (**TV5.6**)
- Provare a modificare un oggetto grafico selezionato (**TV5.7**)
All’utente è chiesto di:
 - Provare a modificare un frame_G (**TV5.8**)
All’utente è chiesto di:
 - * Ridimensionare il frame_G (**TV5.8.1**)
 - * Riposizionare il frame_G (**TV5.8.2**)
 - * Modificare lo stile del frame_G (**TV5.8.3**)
 - Provare a modificare un’area di testo (**TV5.9**)
All’utente è chiesto di:
 - * Ridimensionare l’area di testo (**TV5.9.1**)
 - * Riposizionare l’area di testo (**TV5.9.2**)
 - * Modificare lo stile dell’area di testo (**TV5.9.3**)
 - * Modificare il contenuto dell’area di testo (**TV5.9.4**)
 - * Cambiare il livello dell’area di testo (**TV5.9.5**)
 - Provare a modificare uno shape (**TV5.10**)
All’utente è chiesto di:
 - * Riposizionare lo shape_G (**TV5.10.1**)
 - * Ridimensionare lo shape_G (**TV5.10.2**)
 - * Modificare lo stile dello shape_G (**TV5.10.3**)
 - * Cambiare il livello dello shape_G (**TV5.10.4**)
 - Provare a modificare un’immagine (**TV5.11**)
All’utente è chiesto di:
 - * Riposizionare l’immagine (**TV5.11.1**)
 - * Ridimensionare l’immagine (**TV5.11.2**)
 - * Cambiare il livello dell’immagine (**TV5.11.3**)
- Eliminare un oggetto grafico selezionato (**TV5.12**)
- Creare un nuovo percorso per la presentazione (**TV5.13**)
All’utente è chiesto di:
 - Scegliere un titolo per il percorso creato (**TV5.14**)
 - Clonare un percorso esistente (**TV5.15**)
- Selezionare un percorso tra quelli disponibili (**TV5.16**)
- Modificare il percorso selezionato (**TV5.17**)
All’utente è chiesto di:

- Cambiare il titolo del percorso (**TV5.18**)
- Aggiungere un passo al percorso (**TV5.19**)
- Modificare l'ordine dei frame_G del percorso(**TV5.20**)
- Impostare un frame_G come checkpoint_G (**TV5.21**)
- Rimuovere la marcatura a checkpoint_G da un frame_G (**TV5.22**)
All'utente è chiesto di:
 - * Verificare che si debba confermare l'azione intrapresa (**TV5.22.1**)
 - * Verificare che si possa annullare l'azione intrapresa (**TV5.22.2**)
- Selezionare un frame_G del percorso del percorso (**TV5.23**)
- Eliminare il percorso selezionato (**TV5.24**)
- Modificare il titolo di una presentazione (**TV5.25**)
- Modificare la descrizione di una presentazione (**TV5.26**)

A.3.6 Test TV6

L'utente vuole verificare il corretto funzionamento dell'esportazione
All'utente e` richiesto di:

- Autenticarsi al sistema Premi eseguendo **TV2**
- Selezionare una presentazione dall'elenco delle presentazioni (**TV4.1**)
- Esportare la presentazione come poster
All'utente è chiesto di:
 - Scegliere uno dei formati proposti dal sistema per l'esportazione (**TV6.1**)
- Esportare una presentazione in formato portatile
All'utente è chiesto di:
 - Verificare che il formato prodotto dal sistema sia portatile e quindi eseguibile offline (**TV6.2**)

A.3.7 Test TV7

L'utente vuole verificare il corretto funzionamento della pubblicazione di una presentazione
All'utente e` richiesto di:

- Autenticarsi al sistema Premi eseguendo **TV2**
- Selezionare una presentazione dall'elenco delle presentazioni (**TV4.1**)
- Pubblicare la presentazione
- Verificare che il sistema generi un link alla presentazione e lo renda noto all'utente (**TV7.1**)

A.3.8 Tracciamento Test di Validazione - Requisiti

Requisito	Test di Validazione
Fob13	TV1
Fob14	TV2
FOb16	TV2.1
FOb1	TV3
FOb1.1	TV3.1
FOb1.2	TV3.2
FOb6	TV3
FOb6.1	TV3.3
FOb3	TV4
FOb2	TV4.1
FDe3.1	TV4.2
FOb3.2	TV4.3
FOb3.3	TV4.4
FDe3.4	TV4.5
FOb3.5	TV4.6
FOb3.6	TV4.7
FOb4	TV5
FOb4.1	TV5.1
FOb4.1.1	TV5.2
FOb4.1.1.1	TV5.2.1
FOb4.1.1.2	TV5.2.2
FOb4.1.1.3	TV5.2.3
FOb4.1.1.4	TV5.2.4
FOb4.1.2	TV5.3
FOb4.1.2.1	TV5.3.1
FOb4.1.3	TV5.4
FOb4.1.3.1	TV5.4.1
FOb4.1.4	TV5.5
FOb4.1.4.1	TV5.5.1
FOb4.2	TV5.6
FOb4.3	TV5.7
FOb4.3.1	TV5.8
FOb4.3.1.1	TV5.8.1
FOb4.3.1.2	TV5.8.2
FOb4.3.1.3	TV5.8.3
FOb4.3.3	TV5.9
FOb4.3.3.1	TV5.9.1
FOb4.3.3.2	TV5.9.2
FOb4.3.3.3	TV5.9.3
FOb4.3.3.4	TV5.9.4
FOb4.3.3.5	TV5.9.5
FOb4.3.4	TV5.10
FOb4.3.4.1	TV5.10.1

FOb4.3.4.2	TV5.10.2
FOb4.3.4.3	TV5.10.3
FOb4.3.4.4	TV5.10.4
FOb4.3.2	TV5.11
FOb4.3.2.1	TV5.11.1
FOb4.3.2.2	TV5.11.2
FOb4.3.2.3	TV5.11.3
FOb4.4	TV5.12
FDe4.5	TV5.13
FDe4.5.1	TV5.14
FDe4.5.2	TV5.15
FDe4.6	TV5.16
FOb4.7	TV5.17
FDe4.7.1	TV5.18
FOb4.7.2	TV5.19
FOb4.7.3	TV5.20
FDe4.7.4	TV5.21
FDe4.7.5	TV5.22
FDe4.7.5.1	TV5.22.1
FDe4.7.5.2	TV5.22.2
FOb4.7.6	TV5.23
FDe4.9	TV5.24
FOb4.8	TV5.25
FOb4.10	TV5.26
FOb11	TV6
FOb11.1	TV6.1
FOb11.2	TV6.2
FOb7	TV7
FOb8	TV7.1

B Resoconto dell' Attività di Verifica

In questa sezione vengono descritte le procedure adottate durante il processo di verifica e i risultati ottenuti.

B.1 Revisione della Documentazione

Riguardo all'attività di verifica della documentazione, la checklist stilata dai verificatori durante i controlli sui documenti tramite Inspection è la seguente:

Per i documenti in \LaTeX :

- assenza di doppi spazi;
- uso corretto delle lettere maiuscole e minuscole negli elenchi puntati e all'inizio di ogni frase;
- assenza di errori ortografici di battitura;
- presenza dello spazio dopo il segno di punteggiatura;
- assenza di parti mancanti nei documenti;
- mancanza nel glossario della spiegazione di termini segnati G nei documenti;
- evitare di scrivere frasi troppo lunghe;
- evitare l'inserimento di spazi nei tag \LaTeX ;
- assenza di spazi all'apertura e alla chiusura delle parentesi tonde o quadre;
- presenza dello spazio dopo i segni di punteggiatura;
- verifica del funzionamento dei link dei documenti.

Per i diagrammi UML_G :

- il sistema non deve essere un attore;
- direzione delle frecce scorretta;
- controllo ortografico.

B.2 Tracciamento requisiti

Il tracciamento dei requisiti viene eseguito tramite il software 404TrackerDB descritto nella sezione 2.4.1. Grazie anche allo strumento TexMaker $_G$ descritto nella sezione 2.4.1 si sono potuti individuare errori ortografici mentre la parte di controllo grammaticale è avvenuta mediante la rilettura da parte dei verificatori dei documenti. I verificatori nel segnalare gli errori hanno emesso i ticket ai redattori che sono stati poi risolti dagli stessi. Ciò che si è controllato viene descritto dalla lista seguente:

- ad ogni use case $_G$ deve corrispondere un requisito;

- ad ogni requisito deve corrispondere la sua fonte;
- i requisiti devono coprire l'intero capitolato;
- Ogni requisito deve avere un codice univoco;
- i codici dei casi d'uso nei diagrammi devono corrispondere;
- la numerazione dei casi d'uso non deve contenere salti.

I requisiti presenti nel documento *AnalisiDeiRequisiti_v1.0.pdf* sono:

- **Totali:** 45;
- **Funzionali utente:** 42;
- **di Vincolo:** 3.

Gli use case_G individuati sono 93 tutti lato utente.

B.3 Dettaglio delle verifiche tramite analisi

La verifiche tramite analisi statica avvengono con le modalità walkthrough e inspect spiegate sezione 2.4.2 e permettono di controllare l'andamento e la qualità del lavoro svolto.

B.4 Revisione dei Requisiti

Nel periodo antecedente la consegna di tale revisione sono stati verificati i documenti ed i processi. L'analisi statica e` stata applicata secondo i criteri e le modalità indicate nella sezione 2.5.2. Effettuando walkthrough sono stati riscontrati degli errori. Sono state quindi avviate le procedure per la segnalazione e la correzione, descritte nell'apposita sezione delle *Norme di Progetto v2.0*. Noti gli errori, si e` provveduto a:

- Correggere le imperfezioni rilevate;
- Segnalare gli errori più frequenti. Si e` quindi applicato il ciclo PDCA per rendere più efficiente ed efficace il processo di verifica.

E` stata in seguito applicata l'inspection utilizzando la lista di controllo stilata durante la verifica dei documenti precedentemente verificati, ponendo particolare attenzione ai grafici dei casi d'uso. Il tracciamento (requisiti - fonti, use-case - requisiti) e` stato effettuato tramite l'applicativo 404TrackerDB.

B.5 Revisione di Progettazione

In seguito alle osservazioni effettuate dal committente in sede di RR, il gruppo 404NotFound si è riunito per analizzare l'esito e correggere gli errori rilevati con lo scopo di presentare una versione migliorativa dei vari documenti.

Le principali e più importanti correzioni svolte sono:

- **Analisi dei requisiti:**

- Ristesura dei requisiti secondo le correzioni del committente
- Ristesura dei casi d'uso secondo le correzioni del committente
- Tracciamento ex novo dei Requisiti

- **Piano di Qualifica:**

- Rimossa sezione risorse e spostata nel documento *NormeDiProgetto_v2.0*
- Correzione della sezione 2.3 e introduzione del riferimento alle *Norme di Progetto*
- Stesura appendice A: Pianificazione dei Test
- Descrizione Test di Sistema, Test di Integrazione e Test di Validazione
- Resoconto Revisione di Progettazione

- **Norme di Progetto:**

- Riordinate le sezioni per ambito
- Aggiunte sezioni derivate da altri documenti

- **Piano di Progetto:**

- Cambiate le date di consegna
- Adattati i periodi di durata delle attività
- Ricalcolo dei consuntivi
- Spostate sezioni indicate nelle *Norme di Progetto*

- **Specifica Tecnica:**

- Stesura Integrale del documento

- **Glossario:**

- Cambiamento della struttura del documento
- Aggiunta dell'indice
- Rimozione delle lettere senza definizioni