

13-04-2015



Piano di Qualifica

Informazioni sul documento

Nome Documento	Piano di Qualifica
Versione	1.1.0
Stato	<i>Formale</i>
Uso	<i>Esterno</i>
Data Creazione	20-03-2015
Data Ultima Modifica	13-04-2015
Redazione	Fossa Manuel, Petrucci Mauro
Approvazione	Tollot Pietro
Verifica	Gabelli Pietro
Lista distribuzione	<i>LateButSafe</i> Prof. Tullio Vardanega Prof. Riccardo Cardin Proponente Zucchetti S.p.a.



Tab 1: Versionamento del documento

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

Storico

pre-RR

Versione 1.0	Nominativo
Redazione	Fossa Manuel, Petrucci Mauro
Verifica	Gabelli Pietro
Approvazione	Tollot Pietro

Tab 2: Storico ruoli pre-RR

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	Obiettivi di qualità	7
2.1	Qualità di processo	7
2.2	Qualità di prodotto	9
2.2.1	Funzionalità	9
2.2.2	Affidabilità	9
2.2.3	Efficienza	10
2.2.4	Usabilità	10
2.2.5	Manutenibilità	10
2.2.6	Portabilità	10
2.3	Procedure di controllo di qualità di processo	10
3	Gestione amministrativa della revisione	12
3.1	Comunicazione e risoluzione di anomalie	12
4	Visione generale delle strategie di verifica	13
4.1	Organizzazione	13
4.2	Pianificazione strategica e temporale	13
4.3	Responsabilità	14
4.4	Risorse	14
5	Tecniche e metodi	15
5.1	Tecniche	15
5.2	Metodi	16
5.3	Metriche	16
5.3.1	Metriche _g per la progettazione	17
5.3.2	Metriche _g per il codice	17
5.3.3	Metriche _g per i documenti	18
6	Pianificazione dei test	20
6.1	Test di sistema	20
Appendice A Riassunto delle attività di verifica		29
A.1	Revisione dei Requisiti	29
A.2	Documenti	29

Sommario

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



1.1 Scopo del documento

1.2 Scopo del Prodotto

1.3 Glossario

1.4 Riferimenti

1.4.1 Normativi

- ### 1.4.2 Informativi

- Università degli studi di Padova - 2014/2015

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



2 Obiettivi di qualità

2.1 Qualità di processo

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

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

1. Process performance

Gli indicatori della performance di processo sono:

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

2. Performance Management

L'attuazione di un processo è pianificata e controllata al fine di generare risultati che rispondono agli obiettivi attesi;

3. Work Product Management

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

4. Process Definition

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

5. Process Resource

Il processo può contare su adeguate risorse umane, di infrastrutture, ecc. per essere attuato;

6. Process Measurement

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

7. Process Control

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

8. Process Change

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

9. Continuous Integration

Le modifiche ad un processo sono identificate ed implementate con lo scopo di assicurare il continuo miglioramento nel raggiungere gli obiettivi 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 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 tra le diverse unità operative.

Vi sono poi vari livelli di maturità dei processi che dipendono dal diverso livello di possesso degli attributi:

- **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 approcci sistematici agli 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”.

L'applicazione dello standard ISO/IEC 15504 porta a benefici sia agli sviluppatori del software che ai suoi utilizzatori o acquirenti. Per gli sviluppatori porta vantaggi nell'ottimizzazione dell'uso delle risorse, 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, controllo dello stato di avanzamento in corso d'opera, riduzione dei costi di correzione degli errori ed un controllo dei rischi e delle varianti in corso d'opera.

2.2 Qualità di prodotto

Per garantire la qualità del prodotto si è deciso di seguire le indicazioni fornite dallo standard ISO/IEC 9126:2001 sostituito dal successivo ISO/IEC 25010:2011. Questo documento fornisce un modello per valutare la qualità esterna (nell'ambiente di utilizzo) ed interna (indipendente dall'ambiente) di un software_g, individuando sei caratteristiche principali atte a rendere il prodotto qualitativamente accettabile.

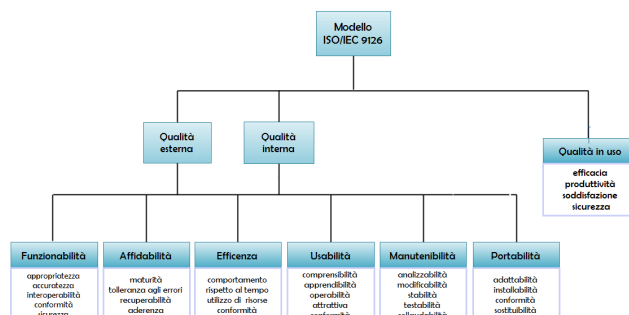


Fig 1: Rappresentazione del modello ISO/IEC 9126:2001

2.2.1 Funzionalità

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

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

2.2.2 Affidabilità

È un requisito_g non funzionale che indica la capacità del software_g di svolgere correttamente il suo compito, mantenendo delle buone prestazioni anche al variare dell'ambiente nel tempo; vengono considerate la sua tolleranza agli errori, la capacità di evitare fallimenti nell'esecuzione a seguito di malfunzionamenti (detta maturità) e la recuperabilità dei dati e delle prestazioni nell'eventualità di un malfunzionamento inevitabile. Il prodotto può considerarsi affidabile se

il numero di esecuzioni andate a buon fine è sufficientemente grande rispetto al numero di esecuzioni totali.

2.2.3 Efficienza

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

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

2.2.4 Usabilità

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

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

2.2.5 Manutenibilità

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

Per misurare questa caratteristica esistono una serie di metriche, descritte nella sezione 5.3.

2.2.6 Portabilità

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

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

2.3 Procedure di controllo di qualità di processo

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



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

3 Gestione amministrativa della revisione

3.1 Comunicazione e risoluzione di anomalie

Un'*anomalia* corrisponde a:

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

In caso un *Verificatore* riscontri un'anomalia, aprirà un ticket_g nel sistema di ticketing_g con le modalità specificate nelle Norme di Progetto_g.

Le modalità di risoluzione di quest'ultimo e la sua struttura vengono descritte in modo dettagliato all'interno del documento [NormeDiProgetto v.1.0.0.pdf](#).

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



4 Visione generale delle strategie di verifica

4.1 Organizzazione

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

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

- Conclusione della prima redazione di un documento;
- Conclusione della prima redazione di un file_g di codice_g;
- Conclusione della modifica sostanziale di un documento: quando il versionamento passa da .x.y.z a .x.y+1.0 oppure a .x+1.0.0. Si veda per approfondimento il paragrafo relativo al versionamento nel documento [NormeDiProgetto_v.1.0.0.pdf](#);
- Conclusione della modifica sostanziale di un file_g di codice_g, quando cioè il versionamento passa da .x.y.z a .x.y +1.0 oppure a .x+1.0.0. Si veda per approfondimento il paragrafo relativo al versionamento nel documento [NormeDiProgetto_v.1.0.0.pdf](#).

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

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

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

$$DIFETTO \longrightarrow ERRORE \longrightarrow FALLIMENTO$$

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_g il ramo di codice_g che le contiene.

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

4.2 Pianificazione strategica e temporale

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

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



la correzione di imprecisioni concettuali e tecniche potrebbe risultare particolarmente gravosa. Come da Piano di Progetto_g di seguito si riportano le quattro milestone_g prima delle quali si effettuerà una verifica del prodotto:

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

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

4.3 Responsabilità

I principali ruoli di responsabilità individuati sono:

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

4.4 Risorse

Per assicurare che gli obiettivi qualitativi vengano raggiunti è necessario l'utilizzo di risorse sia umane che tecnologiche. 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. 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.

5 Tecniche e metodi

5.1 Tecniche

- **Analisi statica:** consiste nell'analizzare il codice_g tramite tools e letture senza tuttavia eseguirlo. Data la natura di questo tipo di analisi, è possibile applicarla anche per il controllo di tutti i documenti testuali prodotti. Si esegue applicando i due seguenti metodi:
 - **Inspection:** l'obiettivo di questa tecnica di analisi è l'individuazione di difetti attraverso la lettura mirata del codice_g. Un prerequisito per questa metodologia di verifica è la definizione di una lista di controllo che elenca le possibili sezioni o passaggi maggiormente soggetti ad errori. La verifica deve essere condotta da soggetti nettamente distinti dai programmatori. La correzione degli errori individuati va eseguita in ogni fase e documentata tramite un rapporto delle attività svolte;
 - **Walkthrough:** l'obiettivo di questa tecnica di analisi è l'individuazione di difetti eseguendo una lettura integrale di tutto il codice_g senza l'assunzione di presupposti. Viene eseguita da gruppi misti di ispettori e sviluppatori. Per evitare incomprensioni, è importante che al termine della lettura gli elementi coinvolti discutano i difetti trovati e che non sia possibile coprire entrambi i ruoli allo stesso tempo. Al termine della fase di discussione si correggeranno i difetti trovati e si apporteranno le modifiche concordate. Anche in questo caso è importante tenere un rapporto delle attività svolte.
- **Analisi dinamica:** consiste nel verificare e validare il software_g o un suo componente osservandone il comportamento in esecuzione durante lo svolgimento di test. Tali test devono essere svolti in maniera ripetibile: significa che se eseguiti nello stesso ambiente e con gli stessi ingressi, devono produrre i medesimi risultati.
 - **Test di unità:** esamina la correttezza di piccole unità di codice_g, generalmente prodotte da un singolo programmatore, in modo da verificare che rispettino i requisiti_g. Può essere svolto con un alto grado di parallelismo servendosi di un automa;
 - **Test di integrazione:** verifica che l'integrazione delle unità che hanno superato il test precedente non produca problemi. Tali problemi, non potendo essere relativi alle singole unità, saranno da ricercare nell'interfaccia che le aggrega;
 - **Test di sistema:** accerta la copertura dei requisiti_g software_g individuati nell'analisi dei requisiti_g permettendo la validazione_g del sistema prodotto;
 - **Test di regressione:** stabilisce se modifiche all'implementazione di un programma_g alterano elementi precedentemente funzionanti. Per far ciò si eseguono nuovamente i test di unità e integrazione sulle parti modificate;
 - **Test funzionali:** mettono alla prova le funzionalità del sistema, simulando l'iterazione_g tra utente e sistema;
 - **Test Prestazionali:** valutano le prestazioni dell'applicazione in molti modi e da molti punti di vista. Questo tipo di test mostra ciò che proverà l'utente in termini

di caricamento e velocità del sito. Le prestazioni sono importanti anche per motivi SEO_g, in quanto un sito lento verrà analizzato molto meno frequentemente dai web_g crawler_g dei motori di ricerca;

- **Test di collaudo:** attività formale supervisionata dal committente_g il cui buon esito comporta la possibilità di rilasciare il prodotto;

5.2 Metodi

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

- **Documenti \LaTeX :**
 1. Rilettura approfondita;
 2. Controllo ortografico tramite lo strumento Aspell;
 3. Controllo dell'applicazione delle regole tipografiche esposte nel documento [Norme-DiProgetto_v.1.0.0.pdf](#);
 4. Verifica della corretta formattazione del file_g pdf prodotto.
- **Codice:** il codice_g verrà analizzato dagli strumenti integrati all'interno dell'IDE_g Aptana e dagli strumenti di sviluppo forniti dai singoli browser_g.
- **Schemi UML:**
 1. data l'impossibilità di controllare la correttezza ortografica degli schemi, con Aspell è necessario esaminare attentamente e più volte i nomi, gli identificativi e i testi nei diagrammi;
 2. controllo della correttezza degli identificativi dei casi d'uso_g rispetto alla nomenclatura stabilita nel documento *Norme di Progetto v.1.0.0*e rispetto alle sezioni dell'Analisi dei Requisiti_g in cui sono inseriti;
 3. controllo della numerazione dei casi d'uso_g rispetto la loro gerarchia;
 4. controllo che i casi d'uso_g soddisfino tutte le esigenze espresse nel capitolato.

5.3 Metriche

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

Una metrica_g è la misura di una proprietà relativa ad una porzione di un documento software_g, allo scopo di fornire informazioni significative sulla qualità del codice_g prodotto.

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

5.3.1 Metriche_g per la progettazione

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

Il valore è calcolato come:

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

5.3.2 Metriche_g per il codice

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

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

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

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

- **Numero linee di codice:** rappresenta il numero di linee di codice_g all'interno di un blocco. Un indice elevato non rappresenta necessariamente un cattivo codice_g ma suggerisce la possibilità di estrarre metodi contenenti gruppi di istruzioni correlate, aumentando il livello di astrazione;
- **Volume di Halstead:** identifica proprietà misurabili del software, e la relazione tra esse. Questi valori sono statisticamente ricavabili dal codice sorgente.

Calcolo:

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

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

Successivamente possono essere calcolati tutti i valori necessari per il Volume di Halstead:

- Program length:

$$N = N1 + N2$$



- Program vocabulary:

$$n = n1 + n2$$

- Volume:

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

- **Indice di manutenibilità:** rappresenta quanto manutenibile è il codice_g, ossia quanto facile è da supportare e migliorare.

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

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

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

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

Parametri utilizzati:

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

5.3.3 Metriche_g per i documenti

- **Indice Gulpease:** misura l'indice di leggibilità di un testo; è tarato sulla lingua italiana. Rispetto ad altri indici ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. Permette di misurare la complessità dello stile di un documento.

L'indice Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

L'indice è calcolato secondo la seguente formula:

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

I risultati sono compresi tra 0 e 100, dove il valore 100 indica la leggibilità più alta e 0 la leggibilità più bassa. In generale risulta che testi con un indice:

- Inferiore a 80 sono difficili da leggere per chi ha la licenza elementare;
- Inferiore a 60 sono difficili da leggere per chi ha la licenza media;
- Inferiore a 40 sono difficili da leggere per chi ha un diploma superiore.

Parametri utilizzati:

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



6 Pianificazione dei test

Si descrivono di seguito tutti i test di validazione, sistema ed integrazione previsti, prevedendo un aggiornamento futuro per i test di unità. Per le tempistiche di esecuzione dei test si faccia riferimento al Piano di Progetto v.2.0.0.

6.1 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 v.2.0.0. I test di sistema riportati sono quelli relativi ai requisiti software individuati e pertanto meritevoli di un test.

Tab 3: 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 3	Viene verificato che ci si possa autenticare con username e password	RF 3
TS 4	Viene verificato che si possa creare una nuova presentazione vuota	RF 4
TS 7	Viene verificato che si possa passare in modalità modifica di una presentazione da desktop _g	RF 7
TS 7.1.1	Viene verificato che si possa inserire un nuovo frame _g nel piano della presentazione _g	RF 7.1.1
TS 7.1.1.1	Viene verificato che si possa scegliere il tipo di frame _g da inserire	RF 7.1.1.1
TS 7.1.4	Viene verificato che si possa spostare un frame _g in modalità modifica	RF 7.1.4
TS 7.1.7	Viene verificato che si possa passare in modalità modifica di un frame _g	RF 7.1.7

Test	Descrizione	Requisito
TS 7.1.7.1	Viene verificato che su possa inserire del testo all'interno di un frame _g	RF 7.1.7.1
TS 7.1.7.4	Viene verificato che si possa modificare del testo già presente all'interno di un frame _g	RF 7.1.7.4
TS 7.1.7.7	Viene verificato che si possa inserire un immagine all'interno del frame _g	RF 7.1.7.7
TS 7.1.7.10	Viene verificato che si possa modificare la dimensione di un immagine	RF 7.1.7.10
TS 7.1.7.13	Viene verificato che si possa inserire un video all'interno di un frame _g	RF 7.1.7.13
TS 7.1.7.16	Viene verificato che si possa modificare la dimensione di un video all'interno di un frame _g	RF 7.1.7.16
TS 7.1.7.19	Viene verificato che si possa spostare un elemento all'interno del frame _g	RF 7.1.7.19
TS 7.1.7.22	Viene verificato che si possa eliminare un elemento presente all'interno del frame _g	RF 7.1.7.22
TS 7.1.7.25	Verificare che si possa modificare il testo di un elemento scelta _g	RF 7.1.7.25
TS 7.1.7.28	Viene verificato che si possa modificare la dimensione di un frame _g	RF 7.1.7.28
TS 7.1.7.31	Viene verificato che si possa modificare la forma di un frame _g	RF 7.1.7.31
TS 7.1.7.34	Viene verificato che si possa modificare lo spessore del bordo di un frame _g	RF 7.1.7.34
TS 7.1.7.37	Viene verificato che si possa modificare il colore del bordo di un frame _g	RF 7.1.7.37
TS 7.1.7.40	Viene verificato che si possa modificare lo sfondo di un frame _g	RF 7.1.7.40, 7.1.7.43

Test	Descrizione	Requisito
TS 7.1.10	Viene verificato che si possa eliminare un frame _g dal piano di una presentazione	RF 7.1.10
TS 7.1.13	Viene verificato che si possa inserire un'immagine di sfondo in un'area della presentazione	RF 7.1.13
TS 7.1.16	Viene verificato che si possa inserire un colore di sfondo in un area della presentazione	RF 7.1.16
TS 7.1.19	Viene verificato che si possa definire un percorso _g di visualizzazione	RF 7.1.19
TS7.1.19.1	Viene verificato che si possa impostare un frame _g iniziale per il percorso _g di presentazione	RF7.1.19.1
TS7.1.19.4	Viene verificato che si possa definire una transizione tra due frame _g	RF7.1.19.4
TS7.1.19.7	Viene verificato che si possa definire una transizione scelta tra due frame _g	RF7.1.19.7
TS7.1.19.10	Viene verificato che si possa eliminare una transizione tra due frame _g	RF7.1.19.10
TS7.1.19.13	Viene verificato che si possa togliere un frame _g dal percorso _g di presentazione	RF7.1.19.13
TS7.1.22	Viene verificato che si possa assegnare un bookmark _g ad un frame _g	RF7.1.22
TS7.1.25	Viene verificato che si possa rimuovere un bookmark _g da un frame _g	RF7.1.25
TS7.1.28	Viene verificato che si possa modificare la velocità di transizione tra due frame _g consecutivi	RF7.1.28
TS7.1.31	Viene verificato che si possa impostare un effetto di transizione tra due frame _g consecutivi	RF7.1.31

Test	Descrizione	Requisito
TS7.1.34	Viene verificato che si possa impostare il tempo di attesa tra due frame _g consecutivi durante la riproduzione automatica	RF7.1.34
TS7.1.37	Viene verificato che si possa annullare e ripristinare una modifica appena effettuata	RF7.1.37, 7.1.40
TS 10	Viene verificato passare in modalità modifica di una presentazione da mobile	RF 10
TS 10.1	Viene verificato che si possa editare testo da mobile all'interno di un frame _g	RF 10.1
TS 10.4	Viene verificato che si possa modificare da mobile il testo presente all'interno di un frame _g	RF 10.4
TS 10.3	Viene verificato che si possa annullare da mobile una modifica appena effettuata	RF 10.3
TS 10.5	Viene verificato che si possa assegnare un bookmark _g ad un frame _g da mobile	RF 10.5
TS 10.8	Viene verificato che si possa rimuovere un bookmark _g ad un frame _g da mobile	RF 10.8
TS 13	Viene verificato che si possa caricare un immagine dal proprio File _g System alla propria parte dedicata alle immagini	RF 13
TS 16	Viene verificato che si possano eliminare dal server _g le immagini caricate	RF 16
TS 19	Viene verificato che si possa creare nuove cartelle e spostare i file _g all'interno delle cartelle	RF 19
TS 25	Viene verificato che si possano organizzare le proprie presentazioni con una struttura a cartelle	RF 25

Test	Descrizione	Requisito
TS 31	Viene verificato che si possano spostare le proprie infografiche _g all'interno della cartella dedicata sul server _g	RF 31
TS 34	Viene verificato che si possa eliminare dal server _g una presentazione creata	RF 34
TS 37	Viene verificato che si possa eliminare dal server _g un'infografica _g creata	RF 37
TS 43	Viene verificato che si possa modificare la propria password di accesso al sistema	RF 43
TS 46	Viene verificato che si possa scaricare in locale un'infografica _g creata sul server _g	RF 46
TS 49	Viene verificato che si possa salvare in locale una presentazione creata sul server _g	RF 49
TS 52	Viene verificato che si possa rimuovere una presentazione salvata in locale	RF 52
TS 55	Viene verificato che si possa eseguire una presentazione salvata sul server _g	RF 55
TS 58	L'utente deve essere in grado di eseguire una presentazione salvata in locale	RF 58
TS 61.1	Viene verificato che si possa eseguire una presentazione in modalità manuale	RF 61.1
TS 61.1.1	Viene verificato che durante la presentazione si possa passare al frame _g successivo o al precedente	RF 61.1.1
TS 61.1.4	Viene verificato che si possa selezionare un elemento scelta _g se presente nel frame _g	RF 61.1.4

Test	Descrizione	Requisito
TS 61.1.7	Viene verificato che si possa passare al frame _g con bookmark _g successivo o precedente	RF 61.1.7
TS 61.1.10	Viene verificato che si possa passare da un frame _g visualizzato al suo frame _g contenitore	RF 61.1.10
TS 61.1.13	Viene verificato che si possa eseguire dello zoom in una parte qualsiasi del frame _g	RF 61.1.13
TS 61.1.16.1	Viene verificato che si possa far partire l'esecuzione di un video all'interno di un frame _g	RF 61.1.16.1
TS 61.1.16.4	Viene verificato che si possa sospendere _g e poi riprendere l'esecuzione di un video all'interno di un frame _g	RF 61.1.16.4
TS 61.1.16.7	Viene verificato che si possa eseguire un video da un punto qualsiasi dello stesso	RF 61.1.16.7
TS 61.1.16.10	Viene verificato che si possa interrompere l'esecuzione di un video	RF 61.1.16.10
TS 61.4	Viene verificato che si possa eseguire una presentazione in modalità automatica	RF 61.4
TS 61.4.1	Viene verificato che si possa chiudere una presentazione in esecuzione automatica	RF 61.4.1
TS 61.4.4	Viene verificato che si possa sospendere _g e riavviare una presentazione in esecuzione automatica	RF 61.4.4
TS 61.4.7	Viene verificato che si possa impostare la velocità di riproduzione della presentazione	RF 61.4.7
TS 61.4.10	Viene verificato che si possa saltare la riproduzione di un video nel frame _g visualizzato	RF 61.4.10

Test	Descrizione	Requisito
TS 70.10.1.4.1	Viene verificato che si possa modificare il font _g del testo	RF 70.10.1.4.1
TS 70.10.1.4.4	Viene verificato che si possa modificare la dimensione del carattere	RF 70.10.1.4.4
TS 70.10.1.4.7	Viene verificato che si possa modificare lo stile del testo	RF 70.10.1.4.7
TS 70.10.1.4.10	Viene verificato che si possa modificare il colore della scritta del testo	RF 70.10.1.4.10
TS 70.10.1.4.13	Viene verificato che si possa modificare il colore dell'evidenziazione del testo	RF 70.10.1.4.13
TS 70.10.1.7	Viene verificato che si possa cambiare la posizione dell'elemento	RF 70.10.1.7
TS 70.10.4	Viene verificato che si possa rimuovere lo sfondo dell'infografica _g	RF 70.4.4
TS 70.10.7	Viene verificato che si possa inserire uno sfondo nell'infografica _g	RF 70.10.7
TS 70.10.10	Viene verificato che si possa inserire un elemento immagine nell'infografica _g	RF 70.10.10
TS 70.10.13	Viene verificato che si possa inserire del testo nell'infografica _g	RF 70.10.13
TS 70.10.16	Viene verificato che si possa inserire un frame _g nella sua interezza presente nella presentazione	RF 70.10.16
TS 70.10.19	Viene verificato che si possano eliminare elementi immagini o testuali	RF 70.10.19
TS 70.13	Viene verificato che si possa salvare l'infografica _g nel suo spazio	RF 70.13
TS 70.16	Viene verificato che si possa annullare e ripristinare una modifica appena effettuata	RF 70.16
TS 70.19	Viene verificato che si possa esportare un'infografica _g in formato stampabile	RF 70.19

Test	Descrizione	Requisito
TS 73	Viene verificato che si possa creare un'infografica _g	RF 73

Tab 4: Descrizione dei test di sistema per i Requisiti di Qualità e Vincoli

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



A Riassunto delle attività di verifica

A.1 Revisione dei Requisiti

Durante questa fase sono stati prodotti solamente documenti di testo quindi sono state applicate le tecniche di analisi statica_g effettuando *walkthrough* e rispettando i metodi definiti nella sezione 5.2.

Nella verifica dei documenti sono stati riscontrati soprattutto errori grammaticali e di battitura dovuti a disattenzioni durante la stesura.

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

A.2 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante la fase di **Analisi**. Un documento è considerato valido soltanto se rispetta le metriche_g descritte su 5.3.3.

Documento	Valore indice	Esito
<i>Piano di Progetto v.1.0.0</i>	89	Superato
<i>Analisi dei Requisiti v.1.0.0</i>	91	Superato
<i>Norme di Progetto v.1.0.0</i>	75	Superato
<i>Piano di Qualifica v.1.0.0</i>	82	Superato
<i>Studio di Fattibilità v.1.0.0</i>	82	Superato
<i>Glossario v.1.0.0</i>	97	Superato

Tab 5: 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.