

NORRIS Framework



FLAMETECH Inc.

Piano di Qualifica

Informazioni sul documento

Versione	1.4.0
Redazione	Meneguzzo Francesco Persegona Mattia Zanetti Davide
Verifica	Cardin Andrea
Responsabile	Sartor Michele
Uso	Esterno
Lista di distribuzione	FlameTech Inc. Prof. Vardanega Tullio

Descrizione

Strategie adottate del gruppo **FlameTech Inc.** per avere un costante controllo dei requisiti qualitativi nello svolgimento del progetto.



Stato	Modifica	Autore	Ruolo	Data	Versione
Approvato	Approvazione documento	Sartor Michele	Responsabile	2015/01/15	1.4.0
Verificato	Verifica documento	Cardin Andrea	Verificatore	2015/01/08	1.3.0
In Lavorazione	Stesura resoconto della verifica	Meneguzzo Francesco	Verificatore	2015/01/07	1.2.0
Verificato	Verifica documento	Meneguzzo Francesco	Verificatore	2014/12/10	1.1.0
In Lavorazione	Stesura sezione sulle metriche	Zanetti Davide	Amminis.	2014/12/07	1.0.3
In Lavorazione	Stesura sezione gestione della revisione	Zanetti Davide	Amminis.	2014/12/05	1.0.2
In Lavorazione	Stesura sezione obiettivi di qualità	Persegona Mattia	Analista	2014/12/03	1.0.1
In Lavorazione	Iniziata stesura documento	Zanetti Davide	Amminis.	2014/12/02	1.0.0

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
2	Obiettivi di qualità	3
2.1	Organizzazione	3
2.2	Pianificazione	3
2.3	Responsabilità	3
2.4	Risorse	3
2.5	Qualità di Processo	4
2.6	Qualità di Prodotto	4
2.7	Controllo qualità di processo	4
2.8	Controllo qualità di prodotto	4
2.9	Tecniche di Analisi	5
2.9.1	Analisi Statica	5
2.9.1.1	Inspection	5
2.9.1.2	Walkthrough	5
2.9.2	Analisi dinamica	6
2.10	Test	6
2.10.1	Test di unità	6
2.10.2	Test di integrazione	6
2.10.3	Test di sistema	6
2.10.4	Test di regressione	7
2.10.5	Test di accettazione	7
3	Misure e Metriche adottate	8
3.1	Metriche per il software	8
3.1.1	Livelli di annidamento	8
3.1.2	Numero parametri per metodo	8
3.1.3	Complessità ciclomatica	8
3.1.4	Linee per metodo	8
3.1.5	Rapporto tra commenti e codice	9
3.1.6	Chiamate innestate di metodi	9
3.1.7	Validazione <i>HTML_G</i>	9
3.1.8	Copertura del codice	9
3.1.9	Numero campi per classe	9
3.2	Metriche per i processi	10
3.2.1	<i>Budget Variance_G</i> (BV)	10
3.2.2	<i>Schedule Variance_G</i> (SV)	10
3.3	Metriche per i documenti	10
3.3.1	Indice di Gulpease _G	10



4	Gestione della revisione	12
4.1	Gestione anomalie riscontrate	12
A	Qualità di processo	13
B	Qualità di Prodotto	16
B.1	Affidabilità	16
B.2	Funzionalità	16
B.3	Manutenibilità	17
B.4	Efficienza	17
B.5	Usabilità	17
B.6	Portabilità	17
B.7	Accoppiamento e Coesione	18
C	Risultati della verifica	19
C.1	Revisione dei Requisiti	19

Elenco delle tabelle

2	Risultati indice di $Gulpease_G$ per la RR	19
---	--	----



Elenco delle figure

1	Modello <i>ISO/IEC 15504_G</i> – <i>SPICE_G</i>	13
2	<i>Ciclo di Deming_G</i> - Plan Do Check Act	15
3	Modello <i>ISO/IEC 9126_G</i>	16

1 Introduzione

1.1 Scopo del documento

Lo scopo del seguente documento è quello di descrivere come il gruppo **FlameTech Inc.** durante tutta la realizzazione del prodotto ha deciso di procedere al fine di ottenere gli obiettivi di qualità di prodotto e di processo prefissati. Per ottenere questi obiettivi si dovrà ricorrere ad una continua verifica delle attività svolte, in modo da poter rilevare immediatamente eventuali errori ed incongruenze commessi. L'identificazione prematura degli errori comporta poi un minor spreco di tempo e di risorse.

1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un *framework_G* per *Node.js_G*, compatibile con l'utilizzo standard dei *middleware_G* di *Express.js_G* in versione 4.x. Il *framework_G* prodotto dovrà permettere la realizzazione rapida di *client_G* web per la visualizzazione di grafici aggiornabili in tempo reale.

1.3 Glossario

Per evitare ogni possibile ambiguità che potrebbe sorgere verrà allegato il *Glossario_ver1.4.0* dove verranno inseriti termini tecnici, acronimi, termini di dominio ed eventuali parole che potrebbero comportare delle incomprensioni e delle ambiguità nella lettura dei documenti. Per rendere la lettura più facile i termini verranno riportati in corsivo ed in pedice verrà posta una "G" maiuscola. (Esempio: *Glossario_G*).

1.4 Riferimenti

1.4.1 Normativi

- *PianoDiProgetto_ver1.4.0*;
- *NormeDiProgetto_ver1.4.0*;
- Capitolato C3
<http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C3.pdf>

1.4.2 Informativi

- Indice di *Gulpease_G*:
http://www.http://it.wikipedia.org/wiki/Indice_Gulpease;
- Slide insegnamento *Ingegneria del Software_G* Mod. A:
<http://www.math.unipd.it/~tullio/IS-1/2014/>;
- Standard ISO_G/IEC_G9126:
http://it.wikipedia.org/wiki/ISO/IEC_9126;
- Standard ISO_G/IEC_G15504:
http://en.wikipedia.org/wiki/ISO/IEC_15504;
- *PDCA_G* – *Ciclo di Deming_G*:
<http://en.wikipedia.org/wiki/PDCA4>;



- Validatore W3C:
<http://validator.w3.org/>.

2 Obiettivi di qualità

2.1 Organizzazione

Il processo di verifica verrà messo in atto ogni qualvolta avverrà un avanzamento di versione, dalla *A.0.C* alla *A.1.0*. Grazie all'ausilio del diario delle modifiche sarà possibile concentrare la propria attenzione solamente nelle sezioni che hanno subito delle modifiche rispetto alla verifica precedente. Come descritto nel *PianoDiProgetto_ver1.4.0*, il progetto è costituito di diverse fasi le quali non potranno essere verificate nello stesso modo a causa della diversità degli output prodotti.

Come conseguenza di questo saranno quindi attuate attività di verifica differenti, specifiche per la fase in cui ci si trova.

- **Fase di Analisi:** devono essere verificati i documenti prodotti ed i processi messi in atto. La descrizione dei metodi di verifica da adottare è descritta nel documento *NormeDiProgetto_ver1.4.0*.

Questa lista verrà aggiornata ad ogni revisione del presente documento, inserendo le parti relative alle fasi completate.

Terminata la fase di verifica si passerà alla fase di approvazione del documento. In questa fase spetterà al *Responsabile di Progetto* assicurarsi che i prodotti realizzati siano conformi a quanto pianificato e progettato.

Il progetto verrà sviluppato seguendo il *modello incrementale_G* di *ciclo di vita_G* del software.

2.2 Pianificazione

Il gruppo **FlameTech Inc.** ha come obiettivo quello di rispettare le scadenze riportate nel *PianoDiProgetto_ver1.4.0*. Per poter adempiere a questo vincolo è necessario che l'attività di verifica del materiale prodotto sia il più possibile sistematica e automatizzata. L'individuazione e la correzione di un errore deve avvenire il prima possibile, per eliminare la possibilità che questo si espanda in maniera esagerata, causando conseguentemente altri errori e portando poi a dover perdere molto tempo per la correzione degli stessi. Per ridurre la possibilità di errori strategici e/o tecnici, ogni stesura del documento sarà preceduta da una attenta analisi e da uno studio preliminare. Attuando questi passi il gruppo prevede di ridurre il numero di interventi correttivi e velocizzare l'attività di verifica.

2.3 Responsabilità

Il processo di verifica è attuato dai membri con il ruolo di *Verificatore*.

La responsabilità di chi ricopre questo ruolo è di garantire che il processo di verifica venga fatto in maniera sistematica e efficace. La descrizione di dettaglio delle attività da svolgere per questi ruoli è specificata nel documento *NormeDiProgetto_ver1.4.0*.

2.4 Risorse

Per realizzare il prodotto con gli obiettivi qualitativi fissati sono necessarie sia risorse umane che tecnologiche. Come detto sopra, le risorse umane con maggiore importanza nelle fasi di verifica sono i membri come il ruolo di *Verificatore*.

Con risorse tecnologiche si intendono invece gli strumenti software che il gruppo ha deciso di adottare per aiutare e velocizzare chi ha il ruolo di *Verificatore* nella fase di verifica. Gli strumenti adottati sono specificati nel documento *NormeDiProgetto_ver1.4.0*.

2.5 Qualità di Processo

Per garantire la qualità del prodotto è necessario garantire la qualità dei processi che lo definiscono. Per mettere in atto questa strategia il gruppo **FlameTech Inc.** ha scelto di adottare lo standard *ISO_G/IEC_G15504*, conosciuto anche con il nome di *SPICE_G*. Questo standard mette a disposizione strumenti utili per valutare la conformità dei processi usati. Al fine di rispettare questo standard verrà utilizzato il *Ciclo di Deming_G*, conosciuto anche come *PDCA_G*. Quest'ultimo mette a disposizione un iter per il controllo dei processi per garantire un miglioramento continuo durante tutto il loro *ciclo di vita_G*.

Maggiori dettagli sono forniti in *Qualità di processo*.

2.6 Qualità di Prodotto

Oltre al controllo sulla qualità di processo, che ha come diretta conseguenza anche l'aumento della qualità del prodotto, verranno garantiti degli appositi controlli sul prodotto stesso, al fine di assicurare che gli obiettivi qualitativi prefissati siano stati raggiunti. Per attuare questa strategia il gruppo farà riferimento allo standard *ISO_G/IEC_G9126*, che ne descrive gli obiettivi e fornisce delle metriche per valutarne il livello di maturità.

Maggiori dettagli sono forniti in *Qualità di Prodotto*.

2.7 Controllo qualità di processo

L'attuazione del *Ciclo di Deming_G* garantirà un costante miglioramento nella qualità di tutti i processi, comportando così anche una maggiore qualità del prodotto. Per poter svolgere con risultati migliori il controllo sui processi, questi devono:

- avere una ripartizione delle risorse chiara ed adeguata;
- essere pianificati in maniera dettagliata;
- essere controllati in modo continuo durante tutto l'arco temporale di svolgimento.

Un prodotto di bassa qualità indica che i processi che lo generano sono migliorabili.

Le metriche adottate per il controllo della qualità sono descritte nella sezione 3.2.

2.8 Controllo qualità di prodotto

Oltre al controllo della qualità del processo verrà anche garantito quello sulla qualità del prodotto. Questo verrà attuato tramite:

- **Verifica:** accerta che l'esecuzione delle attività dei processi svolti non abbia introdotto errori nel prodotto. La verifica verrà svolta durante l'intera fase di realizzazione del prodotto;

- **Validazione:** accerta che il prodotto realizzato sia conforme alle attese e soddisfi i requisiti richiesti;
- **Quality assurance:** per garantire il raggiungimento degli obiettivi di qualità preposti, verranno attuate delle tecniche di analisi statica e dinamica.

Le metriche adottate per il controllo della qualità sono descritte nella sezione 3.1.

2.9 Tecniche di Analisi

Il procedimento di analisi verrà affrontato in 2 forme:

- **Analisi statica:** non richiede l'esecuzione di nessuna parte del prodotto software;
- **Analisi dinamica:** richiede l'esecuzione del software.

2.9.1 Analisi Statica

Questa tecnica di analisi, che non richiede l'esecuzione del software, può essere applicata sia al codice che alla documentazione. La verifica effettuata con questa tecnica porta a rilevare eventuali anomalie ed errori prodotti. Per effettuarla si ricorre a 2 metodi differenti che usualmente vengono applicati in maniera complementare.

2.9.1.1 Inspection

L'attuazione di questa procedura implica una lettura mirata solo di alcune sezioni del documento, cioè quelle che si ritiene siano fonte di maggiori errori, risulta essere quindi molto più rapida del *Walkthrough*.

Essendo eseguita in modo mirato, come prima cosa deve avvenire una pianificazione di quali parti di codice andare a sottoporre alla verifica, poi deve essere definita una *lista di controllo* e solo successivamente si può passare alla lettura vera e propria del documento.

Per questa procedura è richiesta una buona maturità da parte dei verificatori, acquisita nel tempo attraverso la tecnica del *Walkthrough*. Un prerequisito dell'*Inspection* è che le persone che andranno a verificare il documento non siano le stesse che precedentemente lo avevano redatto, ed è necessario che venga presa nota delle modifiche apportate.

2.9.1.2 Walkthrough

Con questo metodo si va ad eseguire invece una lettura a largo spettro di tutto il codice, che di conseguenza impiega più tempo dell'*Inspection*. Questo tipo di controllo viene fatto principalmente nelle prime fasi del progetto, quando l'esperienza dei membri del gruppo non è sufficientemente elevata e quindi c'è bisogno di effettuare una correzione di più ampia portata. Da questa analisi i *Verificatori* sono in grado di capire quali sono gli errori più frequenti e conseguentemente di apportare dei miglioramenti nelle attività per migliorarne le future iterazioni.

Essendo necessario un controllo completo dell'intero documento c'è bisogno di un impiego maggiore di risorse. Terminata la fase di ricerca dei possibili errori sarà effettuata una discussione per esporre i difetti trovati ed organizzare la correzione. Solo al termine di quest'ultima si procederà alla correzione effettiva dei difetti secondo quanto deciso. Anche in questo caso è preferibile che sia redatta una lista delle correzioni apportate.

2.9.2 Analisi dinamica

A differenza dell'analisi statica, l'analisi dinamica viene svolta solamente sul software e necessita della sua esecuzione. L'analisi verrà svolta mediante l'utilizzo di test, appositamente creati, sul codice per individuare possibili malfunzionamenti o errori in fase di progettazione. Perché l'analisi sia efficace c'è bisogno che i test siano ripetibili; questo risulta essere fondamentale perché solo un test che, dato lo stesso input, produce il medesimo output è utile per verificare la corretta implementazione del codice. Per creare dei test ripetibili c'è bisogno di definire a priori i vincoli di:

- **come devono essere eseguiti i test:** bisogna definire l'ordine con il quale i test devono essere eseguiti;
- **relazione tra input ed output:** bisogna definire quali input siano significati per l'esecuzione e quale sia l'output atteso;
- **ambiente d'esecuzione:** definire il sistema hardware e software sul quale andare ad effettuare questi test.

2.10 Test

2.10.1 Test di unità

Nel corso dei test di unità verrà controllata ogni singola componente creata per il prodotto. Questi test verranno eseguiti creando degli appositi *driver_G*, *stub_G* e *logger_G*. Lo scopo è quello di verificare il corretto funzionamento di ogni singola unità che compone il sistema ed eliminare i possibili errori commessi nel corso della programmazione. Durante questi test vengono rilevati circa i $\frac{2}{3}$ dei difetti presenti. Con unità è intesa la più piccola parte di software che è conveniente verificare autonomamente, tipicamente realizzata dallo stesso programmatore. L'individuazione degli errori in questa fase riduce la propagazione degli stessi nei test successivi e comporta una riduzione dei costi, poiché prima viene trovato l'errore e minore è il costo da sostenere per risolverlo.

2.10.2 Test di integrazione

Terminati i test sulle singole unità è possibile procedere all'integrazione delle stesse. La scelta migliore è quella di procedere in maniera incrementale così da avere una base solida funzionante precedentemente verificata. Con questo approccio i possibili malfunzionamenti saranno facilmente tracciabili, in quanto è possibile dimostrare la completa funzionalità fino all'incremento precedente. In questi test potranno emergere errori dovuti a comportamenti non previsti nelle componenti software preesistenti e realizzate da terzi.

Anche durante questi test sarà consigliabile e necessario ricorrere a parti fittizie di utilità, per sostituire le componenti non ancora realizzate.

2.10.3 Test di sistema

Conclusi i test di integrazione si passerà ai test di sistema, i quali consisteranno nella verifica del comportamento dinamico del sistema completo rispetto ai requisiti software. Con questo tipo di test si riscontrerà se il prodotto soddisfa i requisiti software che erano stati stabiliti in fase di *Analisi di Dettaglio*.

2.10.4 Test di regressione

Se una componente, per qualsiasi motivo, dovesse subire delle modifiche sarà necessario andare a rieseguire su di essa i test di unità e di integrazione. Questo è opportuno per accertarsi che la modifica apportata a qualche componente non pregiudichi il funzionamento di altre che non hanno subito modifiche. Un buon tracciamento aiuta a capire quali test di unità e/o integrazione debbano essere ripetuti a causa della modifica apportata.

2.10.5 Test di accettazione

È l'ultimo test, e verrà svolto in presenza del Proponente. L'esito positivo dello stesso comporterà il rilascio ufficiale del software prodotto.

3 Misure e Metriche adottate

Questa sezione contiene tutte le metriche che dovranno essere rispettate nella realizzazione del prodotto.

Per le metriche adottate sarà presente una descrizione, un range di accettazione ed il range ottimale. Il primo rappresenta entro quali valori in prodotto viene accettato, il secondo invece esprime entro quali valori è consigliato rimanere.

Il prodotto non verrà accettato se dovesse sfiorare il range di accettazione.

3.1 Metriche per il software

3.1.1 Livelli di annidamento

Rappresenta quante volte le strutture di controllo sono inserite una all'interno dell'altra. Un valore elevato comporta una difficoltà maggiore in fase di verifica ed un basso livello di astrazione del codice.

- **Range di accettazione:** $1 \rightarrow 6$;
- **Range ottimale:** $1 \rightarrow 3$.

3.1.2 Numero parametri per metodo

Un elevato numero di parametri in input potrebbe portare ad un rapido riempimento dello *stack_G*. Questo può far riflettere sulla possibilità di ridurre il numero di funzionalità per il metodo. Un valore elevato potrebbe essere dovuto ad un errore in fase di progettazione.

- **Range di accettazione:** $0 \rightarrow 8$;
- **Range ottimale:** $0 \rightarrow 4$.

3.1.3 Complessità ciclomatica

La complessità ciclomatica è calcolata utilizzando il grafo del controllo di flusso del programma: i nodi del grafo corrispondono a gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo di istruzioni può essere eseguito immediatamente dopo il primo gruppo. La complessità ciclomatica può, inoltre, essere applicata a singole funzioni, moduli, metodi o classi di un programma. Un valore elevato sta ad indicare una difficile manutenibilità del codice, allo stesso tempo un valore troppo basso potrebbe indicare una scarsa efficienza dei metodi. Questa metrica è utile durante la fase di test, per determinare il numero di casi da realizzare, infatti un valore elevato comporta un numero di test maggiore rispetto ad uno con valore minore.

- **Range di accettazione:** $1 \rightarrow 15$;
- **Range ottimale:** $1 \rightarrow 10$.

3.1.4 Linee per metodo

Rappresenta il numero di *statement_G* che compongono il metodo. Un elevato numero di linee di codice potrebbe renderlo di difficile comprensione e renderne complessa l'eventuale manutenzione. Per ovviare a questo problema potrebbe essere utile dividere lo stesso in più funzioni ed in casi estremi riprogettarne la realizzazione.

- **Range di accettazione:** $0 \rightarrow 50$;
- **Range ottimale** $0 \rightarrow 30$.

3.1.5 Rapporto tra commenti e codice

Rappresenta quante linee di commento sono state scritte rispetto alle linee di codice. La scelta di questa metrica influenza la manutenibilità del codice ed è necessaria a renderlo ben documentato.

$$\frac{\text{linee di commento}}{\text{linee di codice}}$$

- **Range di accettazione:** maggiore 0.25;
- **Range ottimale:** maggiore di 0.35.

3.1.6 Chiamate innestate di metodi

Assieme all'elevato numero di parametri potrebbe portare ad un rapido riempimento dello *stack_G*, la soluzione migliore è quella di limitare il più possibile questo tipo di chiamate.

- **Range di accettazione:** $0 \rightarrow 9$;
- **Range ottimale** $0 \rightarrow 5$.

3.1.7 Validazione *HTML_G*

Il codice *HTML_G* inserito per la creazione della pagina, su cui andare ad inserire i grafici, dovrà superare senza errori il test di validazione fornito dal *W3C_G*. Saranno invece accettati gli avvisi, purché non compromettano la funzionalità e il loro numero sia inferiore ad 8.

- **Range di accettazione:** $0 \rightarrow 8$;
- **Range ottimale** $0 \rightarrow 0$.

3.1.8 Copertura del codice

Indica quante righe di codice sono state eseguite in fase di test a fronte del numero totale realizzato. Più righe di codice vengono coperte minore sarà la probabilità che siano sfuggiti degli errori nelle componenti testate.

- **Range di accettazione:** maggiore del 60%;
- **Range ottimale** maggiore del 75%.

3.1.9 Numero campi per classe

Un alto numero di attributi interni rende la classe poco specializzata e di difficile manutenzione. Questo potrebbe essere la conseguenza di una cattiva progettazione.

- **Range di accettazione:** $0 \rightarrow 15$;
- **Range ottimale** $0 \rightarrow 10$.

3.2 Metriche per i processi

Per monitorare i processi verranno utilizzati degli indici che analizzano i tempi impiegati ed i costi sostenuti.

Gli stessi indici verranno anche usati per avere il controllo sui processi durante il loro svolgimento, questi valori sono anche presenti nel *PianoDiProgetto_ver1.4.0*.

3.2.1 *Budget Variance_G* (BV)

Il BV è utile a monitorare l'aderenza alla pianificazione economica, mettendo in relazione il preventivo pianificato con il consuntivo.

Una $BV > 0$ indica che ci si trova all'interno del budget prefissato e un possibile risparmio economico; per contro, una $BV < 0$ implica un costo superiore a quanto pianificato e la necessità di adeguare il *PianoDiProgetto_ver1.4.0* di conseguenza.

3.2.2 *Schedule Variance_G* (SV)

La SV effettua invece un controllo sull'asse temporale, è utile a capire la differenza tra tempistiche pianificate e reali.

Un valore di $SV > 0$ indica il rispetto delle tempistiche pianificate o in eventuale anticipo delle scadenze, viceversa una $SV < 0$ implica un consumo di tempo maggiore di quanto pianificato.

3.3 Metriche per i documenti

Come analisi sui documenti redatti, verrà verificato l'*indice di Gulpease_G*, che esprime un valore di leggibilità del documento stesso. Per la lingua italiana il numero di possibili indici non è molto elevato, come alternativa è possibile utilizzare l'*indice di Flesch_G*. Il gruppo **FlameTech Inc.**, dopo una valutazione di entrambe le metriche ha deciso di optare per l'utilizzo dell'*indice di Gulpease_G*, in quanto richiede dei valori più facilmente reperibili dai vari strumenti per la redazione di testi, e anche in quanto sono più facilmente reperibili sul web strumenti che offrono la valutazione automatica del testo.

3.3.1 *Indice di Gulpease_G*

L'*Indice di Gulpease_G* è un indice di leggibilità di un testo tarato sulla lingua italiana. Rispetto ad altri ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. L'indice considera due variabili linguistiche:

- la lunghezza della parola;
- la lunghezza della frase rispetto al numero delle lettere.

La formula per il suo calcolo è la seguente:

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

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 possiede la licenza elementare;
- inferiore a 60 sono difficili da leggere per chi possiede la licenza media;
- inferiore a 40 sono difficili da leggere per chi possiede un diploma superiore.

Il gruppo ha stabilito i seguenti range per i propri documenti:

- **Range di accettazione:** 40 \rightarrow 100;
- **Range ottimale:** 50 \rightarrow 100.

4 Gestione della revisione

4.1 Gestione anomalie riscontrate

Se durante la fase di verifica i *Verificatori* dovessero incontrare una o più anomalie, dovranno provvedere ad aprire un ticket usando il sistema di *ticketing_G* scelto. Per i dettagli consultare la sezione 5.2.2 del documento *NormeDiProgetto_ver1.4.0*.

Con anomalie si intende:

- Norme tipografiche non rispettate;
- Errori ortografici;
- Incongruenza del codice con il design del prodotto;
- Incongruenza del prodotto con le funzionalità presenti nell'*Analisi dei Requisiti*;
- Uscita degli indici misurati dai corrispettivi valori di accettazione.

Appendici

A Qualità di processo

Per garantire la qualità del prodotto è necessario, prima, garantire la qualità dei processi che portano alla definizione del prodotto stesso. Per garantire la qualità di processo si è scelto di adottare lo standard *ISO/IEC 15504_G*, conosciuto anche come *SPICE_G*. Questo standard mette a disposizione degli strumenti per valutare il grado di maturità dei processi.

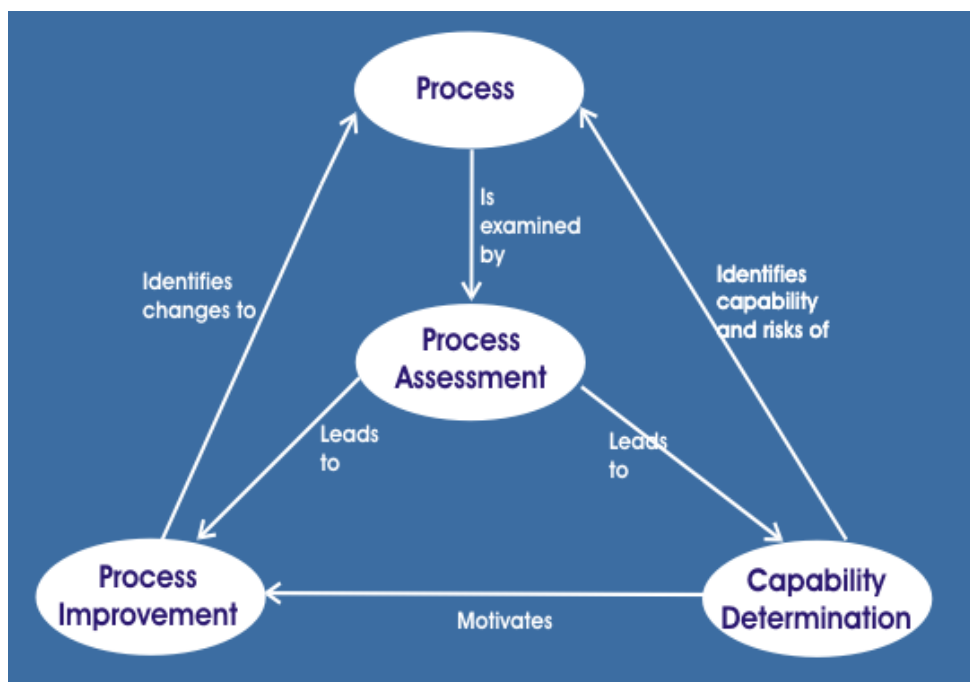


Figura 1: Modello *ISO/IEC 15504_G* – *SPICE_G*

L'idea che sta alla base dello standard è quella che ogni processo debba essere continuamente monitorato per rilevare errori, rischi e anomalie che potrebbero compromettere il raggiungimento degli obiettivi previsti e conseguentemente capirne le cause. Il risultato di ogni singola validazione, per essere attendibile deve poter essere ripetibile. Lo standard definisce 6 livelli di maturità, ognuno possiede degli attributi per determinarla:

0. **Incomplete process:** il processo non viene attuato o non riesce a raggiungere i suoi risultati;
1. **Performed:** il processo attuato raggiunge i suoi risultati.
 - **Process performance attribute:** è la capacità di un processo di raggiungere gli obiettivi, trasformando input identificabili in output identificabili.
2. **Managed process:** il processo viene eseguito in modo controllato secondo obiettivi definiti;
 - **Performance management attribute:** è la capacità del processo di elaborare un prodotto coerente con gli obiettivi fissati;

- **Work product management attribute:** è la capacità del processo di elaborare un prodotto documentato, controllato e verificato.
3. **Established process:** il processo viene eseguito basandosi sui principi dell'ingegneria del software e riesce a raggiungere gli obiettivi fissati;
- **Process definition attribute:** l'esecuzione del processo si basa su standard di processo per raggiungere i propri obiettivi;
 - **Process resource attribute:** è la capacità del processo di attingere a risorse tecniche e umane appropriate per essere attuato efficacemente.
4. **Predictable process:** il processo viene eseguito costantemente entro limiti definiti per raggiungere i risultati attesi;
- **Measurement attribute:** gli obiettivi e le misure di prodotto e di processo vengono usati per garantire il raggiungimento dei traguardi definiti in supporto ai target aziendali;
 - **Process control attribute:** il processo viene controllato tramite misure di prodotto e processo per effettuare correzioni migliorative al processo stesso.
5. **Optimizing process:** il processo cambia e si adatta dinamicamente per raggiungere gli obiettivi aziendali;
- **Process change attribute:** i cambiamenti strutturali, di gestione e d'esecuzione vengono gestiti in maniera controllata allo scopo di raggiungere i risultati fissati;
 - **Continuous improvement attribute:** le modifiche al processo sono identificate e implementate per portare al miglioramento continuo nella realizzazione degli obiettivi di business dell'organizzazione.

Ogni attributo di processo precedentemente descritto è misurabile e lo standard predispone dei differenti livelli:

- **N** non posseduto (0% - 15%);
- **P** parzialmente posseduto (16% - 50%);
- **L** largamente posseduto (51% - 85%);
- **F** completamente posseduto (86% - 100%).

Per permettere l'attuazione dello standard *SPICE* sarà necessario di attuare il principio del *PDCA_G*, conosciuto anche come *Ciclo di Deming_G*:



Figura 2: *Ciclo di Deming*_G - Plan Do Check Act

- **Plan:** stabilire gli obiettivi, le scadenze, le responsabilità e i processi necessari per fornire risultati in accordo con i risultati attesi;
- **Do:** attuare il piano, eseguire il processo e creare il prodotto. Raccogliere poi i dati per la creazione di grafici e analisi;
- **Check:** studiare i risultati effettivi (misurati e raccolti in DO) e confrontarli con i risultati attesi (obiettivi del PLAN) per accertare eventuali differenze;
- **Act:** richiedere azioni correttive sulle differenze significative tra i risultati effettivi e quelli previsti. Analizzare le differenze per determinarne le cause, determinare dove applicare modifiche per ottenere il miglioramento del processo e quindi anche del prodotto.

È stato scelto di avvalersi di questi strumenti così da poter garantire un miglioramento continuo del processo. Per avere sempre sotto controllo la situazione saranno effettuate continue misurazioni sulla qualità e sui miglioramenti apportati. L'attuazione di un continuo miglioramento dei processi avrà come conseguenza un miglioramento della qualità complessiva del prodotto.

B Qualità di Prodotto

Per garantire la qualità del prodotto, il gruppo **FlameTech Inc.** farà riferimento allo standard *ISO/IEC 9126_G*.

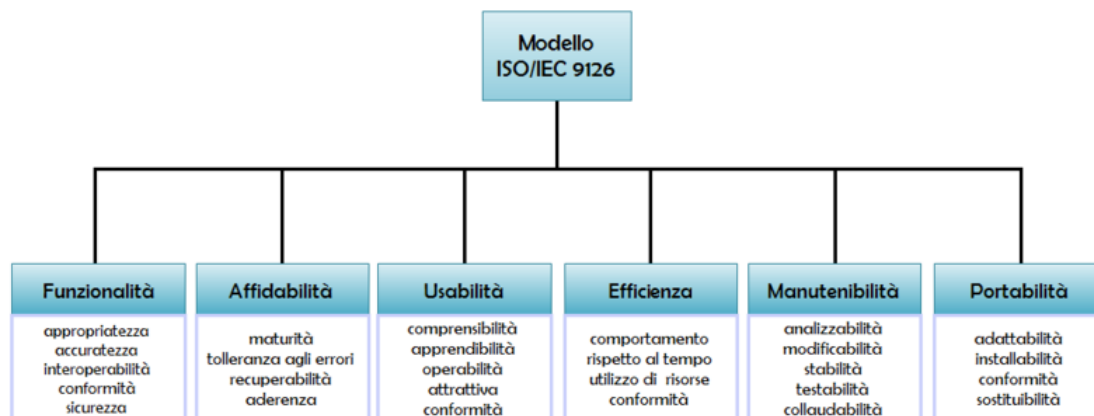


Figura 3: Modello *ISO/IEC 9126_G*

e si impegnerà a garantire le seguenti 6 qualità:

B.1 Affidabilità

Il *framework_G* deve risultare il più robusto possibile, deve avere una buona gestione degli errori che potrebbero sorgere e garantire un adeguato livello di prestazioni.

- **Misura:** l'unità di misura per affermare che il prodotto risulta affidabile sarà la percentuale di test superati, 80% su almeno 20 test effettuati;
- **Metrica:** i test dovranno spaziare il più possibile in tutte le casistiche di cui l'utilizzatore finale potrebbe fare uso, prestando maggiore attenzione ai casi limite;
- **Strumenti:** per soddisfare questa qualità, il sistema deve superare tutti i test previsti.

B.2 Funzionalità

Il *framework_G* prodotto deve garantire tutte le funzionalità indicate nel documento *AnalisiRequisiti_ver1.4.0* e tutte le funzionalità presenti saranno riportate in tale documento. L'implementazione dei requisiti deve essere più completa ed economica possibile.

- **Misura:** l'unità di misura usata sarà la quantità di requisiti mappati in componenti del sistema create e funzionanti;
- **Metrica:** il prodotto si potrà considerare adeguato solo dopo aver dimostrato il soddisfacimento di almeno tutti i requisiti obbligatori;
- **Strumenti:** per verificar questa qualità verranno creati degli appositi test.

Per gli strumenti utilizzati si veda il documento *NormeDiProgetto_ver1.4.0*.

B.3 Manutenibilità

È la capacità del prodotto di essere modificato facilmente in caso di problemi o di revisioni. Per essere manutenibile il prodotto deve essere comprensibile ed estensibile in maniera facile e verificabile.

- **Misura:** per verificare la manutenibilità del software verranno utilizzate le metriche descritte nella sezione *3.1 Metriche per il software*;
- **Metrica:** il software dovrà rispettare tutti i vincoli descritti nella sezione *3.1 Metriche per il software*;
- **Strumenti:** la descrizione degli strumenti è presente nelle *NormeDiProgetto_ver1.4.0*.

B.4 Efficienza

Il software deve avere prestazioni adeguate cercando di ridurre al minimo l'utilizzo delle risorse.

- **Misura:** il tempo impiegato per l'elaborazione e la visualizzazione del grafico nella pagina;
- **Metrica:** il prodotto verrà valutato positivamente se fornirà la richiesta entro massimo 7 secondi;
- **Strumenti:** la descrizione degli strumenti è presente nelle *NormeDiProgetto_ver1.4.0*.

B.5 Usabilità

Il prodotto deve risultare di facile utilizzo, di facile apprendimento e deve rispettare tutte le necessità che un utente potrebbe avere.

- **Misura:** non essendo presenti delle metriche oggettive verrà usata una valutazione soggettiva;
- **Metrica:** per lo stesso motivo starà agli sviluppatori fornire la miglior usabilità possibile.
- **Strumenti:** la descrizione degli strumenti è presente nelle *NormeDiProgetto_ver1.4.0*.

B.6 Portabilità

Il software deve essere in grado di adattarsi, senza troppi oneri, a possibili spostamenti e/o installazioni tra più ambienti di lavoro, anche con caratteristiche diverse.

- **Misura:** il software dovrà essere compatibile almeno con i browser *Chrome* versione 38.x e superiori e *Firefox* dalla versione 32.x;
- **Metrica:** il software dovrà rispettare le metriche descritte nella sezione *3.1 Metriche per il software*;
- **Strumenti:** la descrizione degli strumenti è presente nelle *NormeDiProgetto_ver1.4.0*.

B.7 Accoppiamento e Coesione

Questa caratteristica non rientra tra quelle definite dallo standard *ISO/IEC 9126_G* ma il gruppo **FlameTech Inc.** ritiene sia altrettanto importante.

Il prodotto dovrà avere un basso accoppiamento ed un'alta coesione. Un basso accoppiamento consente di avere una buona comprensione del codice associato ad una classe, senza doversi preoccupare di andare a reperire i dati delle altre classi coinvolte ed eventuali modifiche apportate ad una classe avranno poche o nessuna ripercussione sulle altre che dipendono dalla stessa. Al contrario, un'alta coesione porta ad avere funzionalità vicine nella stessa componente che ne favoriscono la riusabilità ed aiutano a comprendere meglio l'architettura del sistema.

C Risultati della verifica

C.1 Revisione dei Requisiti

Durante le varie attività di realizzazione dei documenti è stata svolta la verifica dei processi e dei documenti realizzati, con particolare attenzione nel periodo precedente alla consegna.

I *Verificatori* hanno analizzato tutti i documenti consegnati tramite *Analisi statica*. La verifica è avvenuta attuando le indicazioni presenti nelle *NormeDiProgetto_ver1.4.0* e utilizzando le metriche descritte nella sezione 3.3 *Metriche per i documenti*. Come primo passo è stato eseguito un *Walkthrough* sui documenti, dal quale sono emersi degli errori. Con questi errori è stata stilata la lista delle anomalie più frequenti. Questa lista verrà utilizzata per controllare le anomalie tramite la tecnica dell'*Inspection*, velocizzando così i tempi delle attività di verifica successive.

Gli errori riscontrati durante il *Walkthrough* sono i seguenti:

- **UML_G:**
 - errori di ortografia;
 - frecce non utilizzate con il significato corretto.
- **Norme stilistiche e L^AT_EX_G:**
 - ruoli di progetto non rappresentati con lo stile corretto;
 - errori di rappresentazione negli elenchi, errati caratteri di terminazione;
 - documenti non scritti utilizzando le macro appositamente realizzate;
 - nome delle persone citate non in forma “Cognome Nome”;
 - parola L^AT_EX_G non viene scritta utilizzando l'apposito comando;
 - errata rappresentazione del formato di data.
- **Errori di ortografia:**
 - frasi di difficile comprensione poiché troppo lunghe;
 - parole non scritte correttamente.

Successivamente all'*Inspection*, per tutti i documenti redatti è stato effettuato, tramite script, il calcolo dell'indice di leggibilità di *Gulpease_G*. I risultati ottenuti sono:

Documento	Valore	Esito
<i>NormeDiProgetto_ver1.4.0</i>	56,75	superato
<i>PianoDiQualifica_ver1.4.0</i>	56,41	superato
<i>AnalisiRequisiti_ver1.4.0</i>	81,99	superato
<i>StudioDiFattibilità_ver1.4.0</i>	56,71	superato
<i>PianoDiProgetto_ver1.4.0</i>	55,99	superato
<i>Glossario_ver1.4.0</i>	53,48	superato

Tabella 2: Risultati indice di *Gulpease_G* per la RR

Dai risultati ottenuti e presenti in tabella tutti i documenti rientrano nel range ottimale che il gruppo aveva prefissato.