

NORRIS Framework



FLAMETECH Inc.

Piano di Qualifica

Informazioni sul documento

Versione	3.0.0
Redazione	Merlo Gianluca Persegona Mattia Zanetti Davide
Verifica	Faggin Andrea
Responsabile	Meneguzzo Francesco
Uso	Esterno
Lista di distribuzione	FlameTech Inc. Prof. Vardanega Tullio Prof. Cardin Riccardo

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	Meneguzzo Francesco	Responsabile	2015/05/19	3.0.0
Verificato	Verifica documento	Faggin Andrea	Verificatore	2015/05/19	2.1.0
In Lavorazione	Terminata stesura appendice D.4 e D.4.1	Merlo Gianluca	Verificatore	2015/05/18	2.0.7
In Lavorazione	Iniziata stesura appendice D.4	Zanetti Davide	Verificatore	2015/05/17	2.0.6
In Lavorazione	Stesura sottosezione Test di Unità in Appendice C	Persegona Mattia	Progettista	2015/05/09	2.0.5
In Lavorazione	Applicazione Correzioni RP - Adeguata appendice A	Zanetti Davide	Amministratore	2015/07/09	2.0.4
In Lavorazione	Applicazione Correzioni RP - Modificato titolo Appendice C	Zanetti Davide	Amministratore	2015/05/08	2.0.3
In Lavorazione	Applicazione Correzioni RP - Rimossa sezione Tecnica di analisi causa contenuti inerenti ad altri documenti	Persegona Mattia	Verificatore	2015/05/08	2.0.2
In Lavorazione	Applicazione Correzioni RP - Rimossa sezione "Gestione della revisione" ed integrata in Appendice D	Persegona Mattia	Verificatore	2015/05/08	2.0.1
Approvato	Approvazione documento	Zanetti Davide	Responsabile	2015/04/10	2.0.0
Verificato	Verifica documento	Faggin Andrea	Verificatore	2015/04/10	1.5.0
In Lavorazione	Stesura appendice D.2	Sartor Michele	Verificatore	2015/04/09	1.4.4
In Lavorazione	Stesura appendice D.1.1	Meneguzzo Francesco	Verificatore	2015/03/10	1.4.3
In Lavorazione	Stesura appendice Pianificazione dei Test	Sartor Michele	Amministratore	2015/03/02	1.4.2
In Lavorazione	Applicazione Correzioni RR - riorganizzazione contenuti	Sartor Michele	Amministratore	2015/02/27	1.4.1

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	Amministratore	2014/12/07	1.0.3
In Lavorazione	Stesura sezione gestione della revisione	Zanetti Davide	Amministratore	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	Amministratore	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	Definizione degli obiettivi	3
2.1.1	Qualità di Processo	3
2.1.2	Qualità di Prodotto	3
2.1.3	Controllo qualità di processo	3
2.2	Controllo qualità di prodotto	3
2.3	Organizzazione	4
2.4	Pianificazione strategica e temporale	5
2.5	Responsabilità	5
2.6	Risorse	5
2.7	Misure e Metriche adottate	5
2.7.1	Metriche per il software	6
2.7.1.1	Livelli di annidamento	6
2.7.1.2	Numero di parametri per metodo	6
2.7.1.3	Complessità ciclomatica	6
2.7.1.4	Linee per metodo	6
2.7.1.5	Rapporto tra commenti e codice	7
2.7.1.6	Chiamate innestate di metodi	7
2.7.1.7	Validazione <i>HTML_G</i>	7
2.7.1.8	Copertura del codice	7
2.7.1.9	Numero campi per classe	7
2.7.2	Metriche per i processi	7
2.7.2.1	<i>Budget Variance_G</i> (BV)	8
2.7.2.2	<i>Schedule Variance_G</i> (SV)	8
2.7.3	Metriche per i documenti	8
2.7.3.1	Indice di <i>Gulpease_G</i>	8
A	Qualità di Processo	10
A.1	Standard <i>ISO/IEC 15504_G</i>	10
A.2	<i>Ciclo di Deming_G</i>	11
B	Qualità di Prodotto	13
B.1	Standard <i>ISO/IEC 9126_G</i>	13
C	Specifica dei Test	16
C.1	Test di integrazione	17
C.1.1	Descrizione dei test di integrazione	17
C.1.2	Tracciamento componenti - test di integrazione	19
C.2	Test di sistema	21
C.2.1	Descrizione dei test di sistema	21



C.3	Test di validazione	23
C.3.1	Test TV1	23
C.3.2	Test TV2	23
C.3.3	Test TV3	23
C.3.4	Test TV4	24
C.3.5	Test TV5	24
C.3.6	Test TV6	24
C.3.7	Test TV7	24
C.3.8	Test TV8	25
C.3.9	Test TV9	26
C.3.10	Test TV10	26
C.4	Test di Unità	27
C.4.1	Tracciamento Test di Unità - Metodi	32
D	Risultati della verifica	35
D.1	Gestione della revisione	35
D.1.1	Gestione anomalie riscontrate	35
D.1.2	Controlli per la qualità di processo	36
D.2	Revisione dei Requisiti	36
D.2.1	Dettaglio esito Revisione dei Requisiti	37
D.3	Revisione di Progettazione	39
D.3.1	Dettaglio esito Revisione di Progettazione	40
D.4	Revisione di Qualifica	41
D.4.1	Misurazioni sul codice	42



Elenco delle tabelle

2	Test di integrazione	19
3	Tabella Tracciamento Componenti - Test di integrazione	20
4	Test di sistema	22
5	Test di Unità	32
6	Tracciamento Test di Unità - metodi	35
7	Risultati indice di BV e SV - Revisione dei Requisiti	37
8	Risultati indice di $Gulpease_G$ - Revisione dei Requisiti	37
9	Risultati indice di BV e SV - Revisione di Progettazione	39
10	Risultati indice di $Gulpease_G$ - Revisione di Progettazione	40
11	Risultati indice di BV e SV - Revisione di Qualifica	41
12	Risultati indice di BV e SV - Revisione di Qualifica	41



Elenco delle figure

1	Modello <i>ISO/IEC 15504_G</i> – <i>SPICE_G</i>	10
2	<i>Ciclo di Deming_G</i> - <i>PDCA_G</i>	12
3	Modello <i>ISO/IEC 9126_G</i>	13
4	<i>V-Model_G</i> per il test software	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_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_ver4.0.0* 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_ver4.0.0*;
- *NormeDiProgetto_ver3.0.0*;
- 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 Definizione degli obiettivi

2.1.1 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 eseguiti. 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 *Appendice A - Qualità di Processo*.

2.1.2 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 *Appendice B - Qualità di Prodotto*.

2.1.3 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.

L'attuazione di tali punti è descritta in *PianoDiProgetto_ver4.0.0*. Un prodotto di bassa qualità indica che i processi che lo generano sono migliorabili. Per aumentare la qualità dei processi questi verranno costantemente monitorati tramite una costante analisi della qualità del prodotto.

Le metriche adottate per il controllo della qualità sono descritte nella sezione 2.7.2.

2.2 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 2.7.1.

2.3 Organizzazione

La strategia di verifica si basa sull'esecuzione di continue verifiche su quanto è stato realizzato. L'attività di verifica viene effettuata su ogni processo implementato, così facendo si ottengono informazioni sulla qualità del processo in esame e conseguentemente anche informazioni sulla qualità del prodotto ottenuto.

Ogni qualvolta viene eseguito un procedimento di verifica, si ottiene un avanzamento di versione del documento in esame, come specificato nelle *NormeDiProgetto_ver3.0.0*. Grazie all'ausilio del diario delle modifiche è quindi possibile concentrare la propria attenzione solamente sulle sezioni che hanno subito delle modifiche rispetto alla precedente attività di verifica. Come descritto nel *PianoDiProgetto_ver4.0.0*, 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.

- **A:** in questa fase vengono redatti i documenti che individuano i requisiti, le strategie e le norme adottate. Si devono adottare i metodi di verifica descritti nelle *NormeDiProgetto_ver3.0.0* sui documenti e sui processi attuati. Il resoconto di tali attività, precedenti la Revisione dei Requisiti, è riportato in appendice *D.2 - Revisione dei Requisiti*;
- **B:** in questa fase vengono verificati i processi che portano all'incremento dei documenti redatti nella fase precedente, e viene controllato che i relativi prodotti seguano quanto descritto nelle *NormeDiProgetto_ver3.0.0*. Il resoconto di tali attività, precedenti la Revisione di Progettazione, è riportato in appendice *D.3 - Revisione di Progettazione*;
- **C:** in questa fase vengono verificati i processi che portano all'incremento dei documenti redatti nella fase precedente, e viene controllato che i relativi prodotti seguano quanto descritto nelle *NormeDiProgetto_ver3.0.0*. Viene verificato inoltre che i prodotti e i processi attuati nell'attività di progettazione dell'architettura seguano le procedure descritte nelle *NormeDiProgetto_ver3.0.0*, e che i test siano pianificati e tracciati in maniera adeguata. Il resoconto di tali attività, precedenti la Revisione di Progettazione, è riportato in appendice *D.3 - Revisione di Progettazione*;
- **D:** in questa fase vengono verificati i processi che portano all'incremento dei documenti redatti nella fase precedente, e viene controllato che i relativi prodotti seguano quanto descritto nelle *NormeDiProgetto_ver3.0.0*. Viene verificato che i prodotti e i processi attuati nelle attività di progettazione di dettaglio e di codifica seguano le procedure descritte nelle *NormeDiProgetto_ver3.0.0*. È inoltre garantito il risultato positivo dei test pianificati nella fase precedente, in conformità con le metriche definite nella sezione 2.7.1: *Metriche per il software*;

- **E:** in questa fase vengono verificati i processi che portano all'incremento dei documenti redatti nella fase precedente, e viene controllato che i relativi prodotti seguano quanto descritto nelle *NormeDiProgetto_ver3.0.0*. Inoltre, viene garantito che il prodotto sia conforme alle attese, e ai requisiti che il gruppo **FlameTech Inc.** si è prefissato di rispettare.

2.4 Pianificazione strategica e temporale

Il gruppo **FlameTech Inc.** ha come obiettivo quello di rispettare le scadenze riportate nel *PianoDiProgetto_ver4.0.0*, e per poter adempiere a questo vincolo è necessario che l'attività di verifica del materiale prodotto sia il più possibile organizzata, 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.

2.5 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 svolto in maniera sistematica ed efficace. La descrizione di dettaglio delle attività da svolgere per questo ruolo è specificata nel documento *NormeDiProgetto_ver3.0.0*.

2.6 Risorse

Per realizzare il prodotto con gli obiettivi qualitativi fissati sono necessarie sia risorse umane che tecnologiche. Come precedentemente specificato, le risorse umane con maggiore importanza nelle fasi di verifica sono i membri che adempiono al 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_ver3.0.0*.

2.7 Misure e Metriche adottate

Poiché il processo di verifica sia informativo deve essere quantificabile. Per farlo è necessario stabilire a priori delle metriche sulle quali basare il processo di verifica. Data la natura incrementale del *ciclo di vita_G* adottato, delle metriche che inizialmente potrebbero risultare incerte e/o approssimate saranno migliorate e definite. Per le metriche adottate sarà presente una descrizione, un range di accettazione ed il range ottimale. Il primo rappresenta entro quali valori il 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.

2.7.1 Metriche per il software

Questa sezione descrive le metriche a cui il gruppo prevede di attenersi per poter perseguire obiettivi di qualità software, ed è da considerarsi indicativa e soggetta a variazioni nelle successive revisioni.

2.7.1.1 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:** $1 \rightarrow 3$.

2.7.1.2 Numero di 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:** $0 \rightarrow 4$.

2.7.1.3 Complessità ciclomatica

La complessità ciclomatica è calcolata utilizzando il grafo del controllo di flusso del programma: i nodi del grafo corrispondono a gruppi atomici 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 implica una difficile manutenibilità del codice, allo stesso tempo un valore troppo basso potrebbe indicare una scarsa efficienza dei metodi. Questa metrica è utile durante l'attività di test, per determinare il numero di casi da realizzare: un valore elevato infatti comporta un numero maggiore di casi da verificare.

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

2.7.1.4 Linee per metodo

Rappresenta il numero di *statement_G* che compongono un 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:** $0 \rightarrow 30$.

2.7.1.5 Rapporto tra commenti e codice

Rappresenta quante linee di commento sono state scritte rispetto alle linee di codice. Questo valore influenza la leggibilità e la manutenibilità del codice.

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

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

2.7.1.6 Chiamate innestate di metodi

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

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

2.7.1.7 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:** $0 \rightarrow 0$.

2.7.1.8 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%.

2.7.1.9 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:** $0 \rightarrow 10$.

2.7.2 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 controllo sui processi durante il loro svolgimento, questi valori sono anche presenti nel documento *PianoDiProgetto_ver4.0.0*.

2.7.2.1 *Budget Variance_G* (BV)

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

$$BV = PV - AC$$

Dove *PV* (*Planned Value*) rappresenta il budget pianificato per l'attività mentre *AC* (*Actual Cost*) rappresenta quanto effettivamente speso. Un $BV > 0$ indica che ci si trova all'interno del budget prefissato e un possibile risparmio economico; per contro, un $BV < 0$ implica un costo superiore a quanto pianificato e la necessità di adeguare il documento *PianoDiProgetto_ver4.0.0* di conseguenza.

- **Range di accettazione:** $BV \leq PV * 1.1\%$;
- **Range ottimale:** $BV \geq 0$.

2.7.2.2 *Schedule Variance_G* (SV)

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

$$SV = EV - PV$$

Dove *EV* (*Earned Value*) rappresenta il valore delle attività realizzate nella data in esame mentre *PV* (*Planned Value*) rappresenta il costo pianificato per la stessa data. Un valore di $SV < 0$ indica il rispetto delle tempistiche pianificate o eventuale anticipo delle scadenze, viceversa una $SV > 0$ implica un consumo di tempo maggiore di quanto pianificato.

- **Range di accettazione:** $SV \leq EV * 1.05\%$;
- **Range ottimale:** $EV \geq 0$.

2.7.3 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.

2.7.3.1 Indice di *Gulpease_G*

Rispetto ad altre metriche, l'indice di *Gulpease_G* ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. L'indice considera due variabili linguistiche:

- lunghezza della parola;
- 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 → 100;
- **Range ottimale:** 50 → 100.

Appendici

A Qualità di Processo

A.1 Standard *ISO/IEC 15504_G*

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*. Questo standard mette a disposizione degli strumenti per valutare il grado di maturità dei processi.

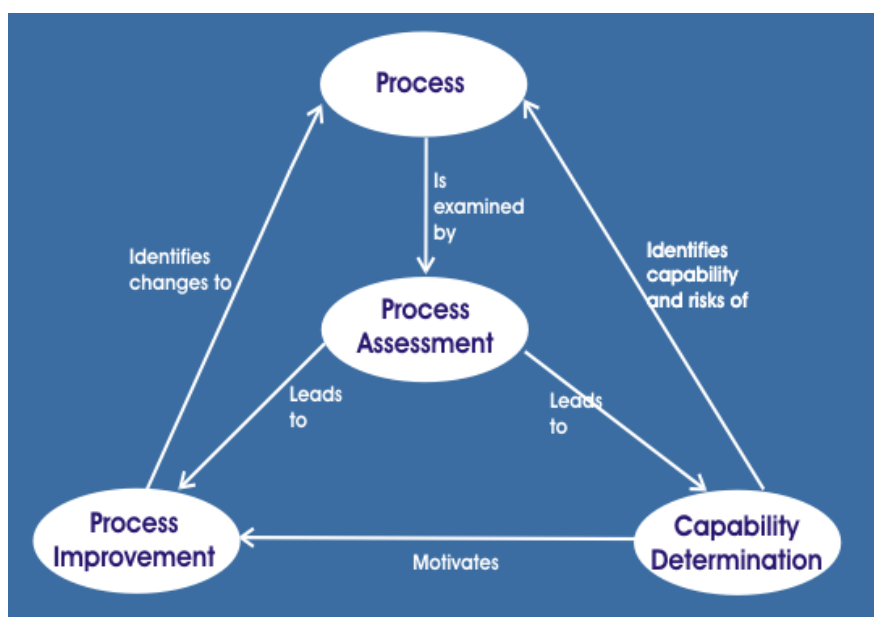


Figura 1: Modello *ISO/IEC 15504_G* – *SPICE_G*

L'idea alla base dello standard è quella che ogni processo debba essere continuamente monitorato per rilevare errori e rischi a esso intrinseci che potrebbero compromettere il raggiungimento degli obiettivi previsti, in modo da migliorarne l'efficienza. Il risultato di ogni singola valutazione, per essere attendibile deve poter essere ripetibile. Lo standard definisce 6 livelli di maturità, ognuno dei quali con degli attributi per determinarla:

1. **Incomplete process:** il processo non viene attuato o non riesce a raggiungere i suoi risultati;
2. **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.
3. **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.
4. **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.
5. **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.
6. **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%).

A.2 *Ciclo di Deming*

Il $PDCA_G$ è uno strumento di miglioramento continuo del sistema aziendale. È uno dei principali strumenti per il miglioramento della qualità dei processi. $PDCA_G$ è l'acronimo di Plan - Do - Check - Act, il metodo è anche conosciuto come *ciclo di Deming* dal nome del suo ideatore, W. Edwards Deming. Il metodo si divide in 4 fasi:



Figura 2: *Ciclo di Deming_G - PDCA_G*

- **Plan**(pianificazione): il miglioramento deve essere progettato e pianificato. È necessario conoscere gli obiettivi e i soggetti del sistema, le criticità e le priorità dell'impresa. Nella fase di pianificazione si analizza la situazione attuale (criticità, opportunità, punti di forza, punti di debolezza), si elabora un piano di azione e si descrivono le attese di miglioramento;
- **Do**(esecuzione): in questa fase si passa all'attuazione delle azioni pianificate. In questa fase (Do) si rendono operative le decisioni prese nella fase di pianificazione (Plan);
- **Check**(controllo): fase di controllo dei risultati raggiunti. Si analizzano i feedback. Viene comparata la situazione corrente con la situazione iniziale. Si quantificano i risultati ottenuti per misurare gli effetti positivi del piano d'azione, verificando che i risultati ottenuti siano effettivamente determinati dall'esecuzione (Do) piano di azione (Plan). I risultati possono essere sia quantitativi che qualitativi;
- **Act**(azione): se l'obiettivo di miglioramento atteso viene raggiunto, si deve cercare di standardizzare le azioni intraprese e rendere stabile la situazione e i risultati ottenuti. Viceversa, se l'obiettivo di miglioramento atteso non viene raggiunto, si riesamina il ciclo Plan-Do alla ricerca di errori di attuazione, al fine di elaborare un nuovo ciclo *PDCA_G*.

B Qualità di Prodotto

B.1 Standard *ISO/IEC 9126_G*

Per garantire il rispetto di obiettivi qualitativi di prodotto, il gruppo **FlameTech Inc.** farà riferimento allo standard *ISO/IEC 9126_G*, con lo scopo di misurare il raggiungimento di tali obiettivi.

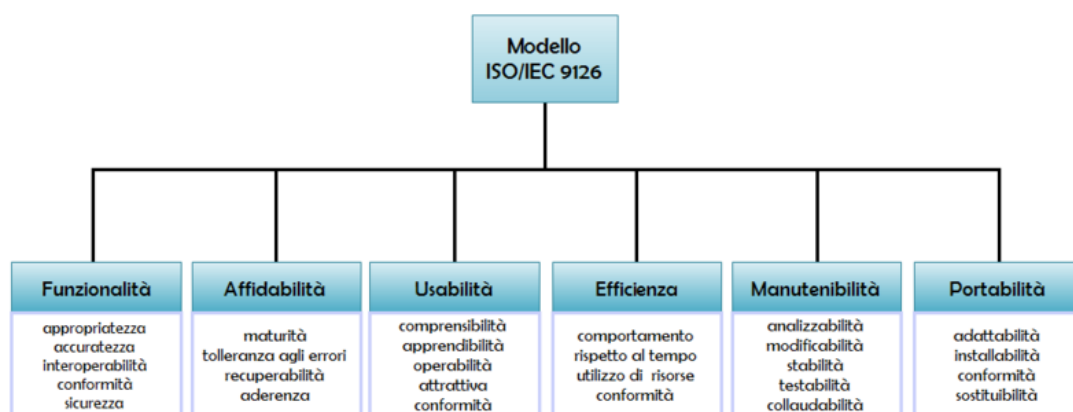


Figura 3: Modello *ISO/IEC 9126_G*

Lo standard divide gli obiettivi qualitativi in tre categorie: Qualità in uso, obiettivi di Qualità esterna e di Qualità interna. Non essendo valutabile la qualità in uso di un software in corso di sviluppo, il gruppo ha deciso di rispettare le sei caratteristiche qualitative principali di qualità esterna ed interna, suddivise in ulteriori sotto caratteristiche che possono essere misurate:

- **Reliability:** è la capacità del software di essere il più robusto possibile e di garantire un adeguato livello di prestazioni nel tempo;
 - **Maturity:** capacità del prodotto software di evitare errori, malfunzionamenti o risultati non corretti dovuti ad errori nel prodotto;
 - **Fault tolerance:** capacità del prodotto di mantenere livelli adeguati di prestazioni anche in presenza di malfunzionamenti e/o usi scorretti o non previsti delle funzionalità;
 - **Recoverability:** capacità del prodotto di ripristinare il livello appropriato di prestazioni e di recuperare informazioni rilevanti in seguito ad un malfunzionamento;
 - **Reliability Compliance:** capacità del software di aderire a standard, regole e convenzioni inerenti all'affidabilità.
- **Functionality:** è la capacità del prodotto software di fornire funzioni che soddisfano esigenze stabilite;
 - **Suitability:** capacità del prodotto software di fornire un appropriato insieme di funzioni per le attività specifiche;
 - **Accuracy:** capacità del prodotto software di fornire i risultati previsti;
 - **Interoperability:** capacità del prodotto software di interagire ed operare con uno o più sistemi specifici;

- **Security:** capacità del prodotto software di proteggere informazioni e dati, evitando accessi non autorizzati e garantendo accessi autorizzati;
- **Functionality Compliance:** capacità del prodotto software di aderire a standard, e regolamenti in materia di funzionalità.
- **Maintainability:** è la capacità del software di essere modificato posteriormente al rilascio, includendo correzioni, miglioramenti o adattamenti;
 - **Analyzability:** capacità del prodotto software di poter essere analizzato per localizzare un errore nello stesso;
 - **Changeability:** capacità del prodotto software di permettere l'implementazione di una data modifica;
 - **Stability:** capacità del prodotto software di evitare effetti inaspettati derivanti da modifiche effettuate;
 - **Testability:** capacità del prodotto software di essere facilmente testato per validare le modifiche apportate.
- **Efficiency:** è la capacità del prodotto di fornire prestazioni in relazione alla quantità di risorse usate;
 - **Time Behaviour:** capacità del prodotto software di fornire adeguati tempi di risposta e di elaborazione;
 - **Resource Utilization:** capacità del prodotto software di utilizzare quantità e tipologie di risorse differenti in maniera adeguata;
 - **Efficiency Compliance:** capacità del prodotto software di aderire a standard e specifiche di efficienza.
- **Usability:** è la capacità del prodotto software di essere appreso e usato dall'utente, nonché attraente per quest'ultimo;
 - **Understandability:** capacità del prodotto software di rendere comprensibili i concetti espressi, mettendo l'utente in condizione di capire se è appropriato;
 - **Learnability:** capacità del prodotto software di ridurre l'impegno richiesto agli utenti per impararne le funzionalità espresse;
 - **Operability:** capacità del prodotto software di mettere l'utente in condizione di farne uso per i propri scopi;
 - **Attractiveness:** capacità del prodotto software di essere piacevole e interessante per l'utente;
 - **Usability Compliance:** capacità del prodotto software di aderire a standard, e regolamenti in materia di usabilità.
- **Portability:** è la capacità del prodotto software di essere trasportato da un ambiente di lavoro ad un altro;
 - **Adaptability:** capacità del prodotto software di essere adattato per diversi ambienti operativi senza dover effettuare modifiche diverse da quelle fornite;
 - **Installability:** capacità del prodotto software di essere installato in uno specificato ambiente;

- **Co-Existence:** capacità del prodotto software di coesistere con altri software, con ambienti e risorse comuni;
- **Replaceabilty:** capacità del prodotto software di essere utilizzato in sostituzione di un altro per svolgere gli stessi compiti nello stesso ambiente;
- **Portabilty Compliance:** capacità del prodotto software di aderire a standard, e regolamenti in materia di portabilità.

C Specifica dei Test

Di seguito sono descritti i test di integrazione, sistema e validazione che verranno eseguiti. È invece previsto un aggiornamento futuro per la descrizione dei test di unità. Le informazioni sulle tempistiche di esecuzione invece sono descritte nel *PianoDiProgetto_ver4.0.0*.

Nelle tabelle seguenti, che riporteranno i test suddividendoli per tipologia, sarà possibile trovare nella colonna "Stato" la dicitura **NT**, ad indicare che il test non è ancora stato eseguito e verrà effettuato in un momento successivo.

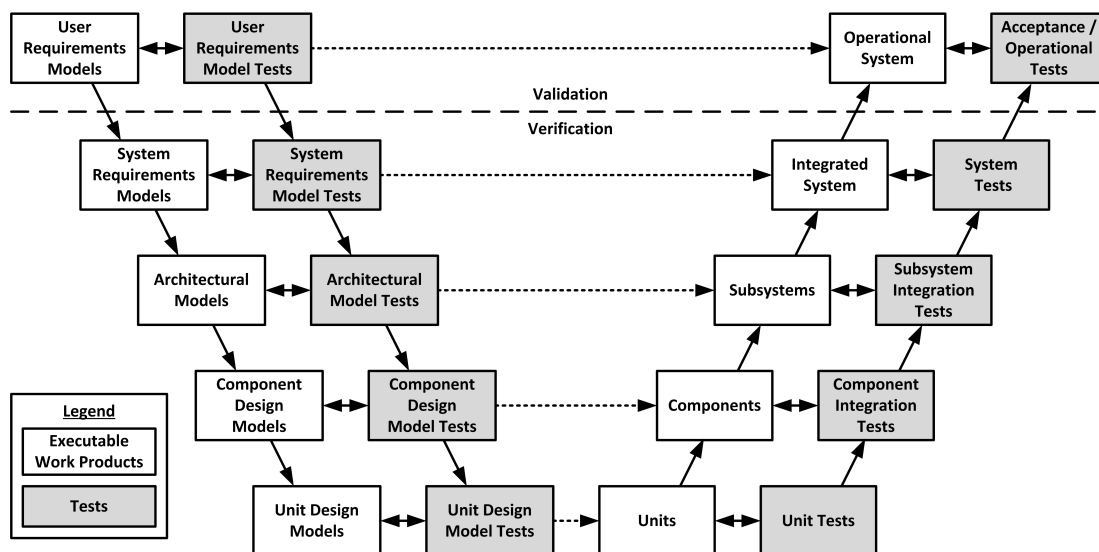


Figura 4: *V-Model_G* per il test software

I test saranno identificati dalla seguente codifica:

- test di unità: **TU**[Codice progressivo test]
- test di integrazione: **TI**.[Nome del componente]
- test di sistema: **TS**[Codice requisito]
- test di validazione: **TV**[Codice progressivo test]

C.1 Test di integrazione

In questa sezione vengono descritti i test di integrazione eseguiti sui vari componenti descritti nella progettazione di alto livello. Questi test vengono eseguiti per verificare la corretta integrazione ed il corretto flusso dei dati all'interno del sistema.

Si è scelto di adottare un approccio di integrazione incrementale, così facendo sarà possibile sviluppare e verificare più componenti parallelamente.

Assemblando le componenti in modo incrementale i possibili difetti che emergono da un test saranno da attribuire, con grande probabilità, all'ultima componente integrata. Questa tecnica permette di rendere reversibile ogni passo, facilitando il ritorno a uno stato di cui è noto il funzionamento.

Si è inoltre deciso di utilizzare il metodo *bottom-up_G*, così da poter integrare prima le parti con minore dipendenza funzionale e maggiore funzionalità, corrispondenti alle componenti collegate ai requisiti obbligatori, permettendo quindi di avere una versione funzionante di queste componenti in tempi rapidi. In questo modo, peraltro, le componenti obbligatorie saranno testate più volte, riducendo così la possibile presenza di errori. Successivamente si risalirà l'albero delle dipendenze per arrivare alle componenti di alto livello sulle quali saranno eseguiti gli ultimi test.

C.1.1 Descrizione dei test di integrazione

Test	Descrizione	Stato	Componente
TI.Norris	Verificare che le componenti Back-end e Front-end si integrino correttamente tra loro garantendo un corretto funzionamento complessivo del sistema	NT	Norris
TI.Lib	Verificare che la libreria del frameworkG Norris importi automaticamente i moduli necessari al proprio funzionamento e che il BusinessLayer, il DataLayer e il PresentationLayer si integrino correttamente tra loro per un corretto funzionamento	NT	Norris:: Lib
TI.BusinessLayer	Verificare che venga gestito correttamente il componente BusinessLayer relativo al <i>Three Tier Architecture</i> , in particolare che venga gestita correttamente la comunicazione tra le componenti PresentationLayer e DataLayer	NT	Norris:: Lib:: BusinessLayer

Test	Descrizione	Stato	Componente
TI.DataLayer	Verificare che venga gestito correttamente il componente DataLayer relativo al <i>Three Tier Architecture_G</i> , in particolare che vengano gestite correttamente le informazioni relative alle pagine ed ai grafici	NT	Norris:: Lib:: DataLayer
TI.PresentationLayer	Verificare che venga gestito correttamente il componente PresentationLayer relativo al <i>Three Tier Architecture_G</i> , in particolare che vengano gestite correttamente le richieste <i>HTTP_G</i> effettuate dai <i>client_G</i>	NT	Norris:: Lib:: PresentationLayer
TI.NorrisApp	Verificare che le componenti Model, View, Controller e Services si integrino correttamente tra loro permettendo la corretta visualizzazione di pagine e grafici ed il loro aggiornamento	NT	Norris:: NorrisApp
TI.Controllers	Verificare che venga gestito correttamente il componente Controller relativo al <i>design pattern_G MVC_G</i> , in particolare che vengano creati correttamente i grafici e che la pagina visualizzata dal <i>client_G</i> sia gestita correttamente	NT	Norris:: NorrisApp:: Controllers
TI.Model	Verificare che venga gestito correttamente il componente Model relativo al <i>design pattern_G MVW_G</i> , in particolare che vengano mantenuti e gestiti correttamente i dati dei grafici presenti sulla pagina visualizzata dal <i>client_G</i>	NT	Norris:: NorrisApp:: Model
TI.Services	Verificare che venga gestito correttamente il componente Service relativo al <i>design pattern_G MVW_G</i> , in particolare che vengano gestite correttamente le notifiche di aggiornamento da parte del <i>server_G</i>	NT	Norris:: NorrisApp:: Services

Test	Descrizione	Stato	Componente
TI.Views	Verificare che venga gestito correttamente il componente View relativo al <i>design pattern_G MVW_G</i> , in particolare che il sistema gestisca correttamente i template delle pagine <i>HTML_G</i> e il codice <i>JavaScript_G</i>	NT	Norris:: NorrisApp:: Views
TI.AndroidApp	Verificare che le componenti Layouts, Activities e AppModel si integrino correttamente tra loro garantendo un corretto funzionamento complessivo dell'applicativo <i>Android_G</i>	NT	AndroidApp
TI.Activities	Verificare che venga gestito correttamente il componente Controller relativo al <i>design pattern_G MVC_G</i> , in particolare che venga gestita correttamente la logica delle varie attività dell'applicativo	NT	AndroidApp:: Activities
TI.AppModel	Verificare che venga gestito correttamente il componente Model relativo al <i>design pattern_G MVC_G</i> , in particolare che vengano mantenuti e gestiti correttamente i dati necessari al funzionamento dell'applicativo	NT	AndroidApp:: AppModel
TI.Layouts	Verificare che venga gestito correttamente il componente View relativo al <i>design pattern_G MVC_G</i> , in particolare che vengano visualizzati i layout corretti per le varie funzionalità dell'applicativo	NT	AndroidApp:: Layouts

Tabella 2: Test di integrazione

C.1.2 Tracciamento componenti - test di integrazione

Componente	Test integrazione
Norris	TI.Norris
Norris::Lib	TI.Lib
Norris::Lib::BusinessLayer	TI.BusinessLayer

Componente	Test integrazione
Norris::Lib::DataLayer	TI.DataLayer
Norris::Lib::PresentationLayer	TI.PresentationLayer
Norris::Lib::Utils	
Norris::NorrisApp	TI.NorrisApp
Norris::NorrisApp::Controllers	TI.Controllers
Norris::NorrisApp::Model	TI.Model
Norris::NorrisApp::Services	TI.Services
Norris::NorrisApp::Views	TI.Views
AndroidApp	TI.AndroidApp
AndroidApp::Activities	TI.Activities
AndroidApp::AppModel	TI.AppModel
AndroidApp::Layouts	TI.Layouts

Tabella 3: Tabella Tracciamento Componenti - Test di integrazione

C.2 Test di sistema

In questa sezione sono descritti i test di sistema che permettono di verificare il comportamento dinamico del sistema nella sua interezza, rispetto ai requisiti software individuati nel documento *AnalisiRequisiti_ver4.0.0*.

C.2.1 Descrizione dei test di sistema

Test	Descrizione	Stato	Requisito
TSF1	Verificare che il <i>framework_G</i> renda disponibili delle <i>API_G</i> per la realizzazione di grafici e per l'accorpamento di tali grafici in pagine	NT	RAF1
TSF1.1	Verificare che il sistema notifichi correttamente le modifiche apportate ad un grafico a tutti i <i>client_G</i> connessi	NT	RAF1.1
TSF1.2	Verificare che il sistema aggiorni correttamente i dati di un grafico in seguito ad una modifica	NT	RAF1.2
TSF1.3	Verificare che il sistema permetta di aggiungere un grafico alla pagina	NT	RAF1.3
TSF2	Verificare che il sistema permetta di configurare vari parametri sulle pagine create	NT	RAF2
TSF3	Verificare che sia possibile creare grafici di tipo <i>Bar Chart</i> , <i>Line Chart</i> , <i>Map Chart</i> e <i>Table</i>	NT	RAF3
TSF3.5	Verificare che i grafici creati siano visualizzati correttamente	NT	RAF3.5
TSF4	Verificare che il <i>framework_G</i> permetta di configurare vari parametri sui grafici	NT	RAF4
TSF5	Verificare che sia disponibile una <i>dashboard_G</i> per la visualizzazione degli spostamenti degli autobus della società APS Holding nella città di Padova	NT	RAF5
TSQ1	Verificare che per lo sviluppo del prodotto siano state rispettate le norme descritte nel documento <i>NormeDiProgetto_ver3.0.0</i>	NT	RAQ1
TSQ2	Verificare che Norris <i>framework_G</i> sia distribuito con <i>Licenza MIT_G</i> e che menzioni il contributo di CoffeeStrap	NT	RAQ2

Test	Descrizione	Stato	Requisito
TSQ3	Verificare che venga fornito il manuale utente in lingua italiana	NT	RAQ3
TSQ3.1	Verificare che venga fornito il manuale utente in lingua inglese	NT	RBQ3.1
TSQ4	Verificare che venga fornita una guida per l'utilizzo dell'applicazione <i>Android_G</i>	NT	RAQ4
TSV1	Verificare che la componente <i>server_G</i> sia stata realizzata utilizzando <i>Node.js_G</i>	NT	RAV1
TSV2	Verificare che il sistema utilizzi la versione 4.x di <i>Express_G</i> per la realizzazione dell'infrastruttura per la componente <i>server_G</i>	NT	RAV2
TSV3	Verificare che il sistema utilizzi <i>Socket.io_G</i> per la componente <i>WebSocket_G</i> che realizza le notifiche <i>push_G</i>	NT	RAV3
TSV4	Verificare che il sistema lato <i>client_G</i> sia completamente compatibile con il browser <i>Chrome_G</i> versione 38.0.X o superiori	NT	RAV4
TSV5	Verificare che il sistema lato <i>client_G</i> sia completamente compatibile con il browser <i>Firefox_G</i> versione 32.X o superiori	NT	RAV5
TSV6	Verificare che il progetto sia caricato su <i>GitHub_G</i> e permetta l'utilizzo delle <i>issue_G</i> per la segnalazione di <i>bug_G</i>	NT	RAV6
TSV7	Verificare che sia stato effettuato il <i>deployment_G</i> su <i>Heroku_G</i>	NT	RCV7
TSV8	Verificare che sia possibile installare il <i>framework_G</i> tramite l'utilizzo del <i>repository_G npm_G</i> di <i>Node.js_G</i>	NT	RAV8
TSV8.1	Verificare che il <i>framework_G</i> importi automaticamente le librerie esterne necessarie al funzionamento quali <i>Express_G</i> e <i>Socket.io_G</i>	NT	RBV8.1
TSV9	Verificare che l'applicazione <i>Android_G</i> sia compatibile con la versione 4.1 o superiori di <i>Android_G</i>	NT	RAV9

Tabella 4: Test di sistema

C.3 Test di validazione

In questa sezione sono descritti i test di validazione, necessari per verificare che il prodotto realizzato sia effettivamente conforme alle attese. Per ogni test saranno descritti i passi che l'utente dovrà eseguire per testare i requisiti collegati. Per esaminare il tracciamento di questi test con i requisiti corrispondenti si rimanda al documento *AnalisiRequisiti_ver4.0.0*. A causa della presenza di vari step da eseguire per l'esecuzione del test l'utilizzo di una rappresentazione tabellare risultava poco chiara e di difficile comprensione. A seguito di questo si è scelto di presentarli sotto forma di lista.

C.3.1 Test TV1

Lo sviluppatore vuole verificare che le API_G disponibili permettano la realizzazione di grafici e l'accorpamento degli stessi in pagine.

Allo sviluppatore è richiesto di:

- Seguire la procedura per poter utilizzare le API_G rese disponibili (TV1.1);
- Creare una pagina, aggiungervi un grafico e verificare che sia possibile assegnargli le dimensioni di visualizzazione (TV1.2);
- Verificare che le API_G permettano la creazione di grafici ed il loro accorpamento in pagine (TV1.3).

C.3.2 Test TV2

Lo sviluppatore vuole verificare che il $framework_G$ aggiorni tramite notifiche $push_G$ i grafici sui $client_G$ connessi per ogni modifica apportata.

Allo sviluppatore è richiesto di:

- Verificare che il $framework_G$ permetta di aggiornare i dati dei grafici. (TV2.1)
Allo sviluppatore è richiesto di:
 - Verificare che il $framework_G$ permetta l'aggiornamento in place per tutti i tipi di grafico disponibili (TV2.1.1);
 - Verificare che il $framework_G$ permetta l'aggiornamento stream per i grafici di tipo *Line Chart_G* e *Table_G* (TV2.1.2);
 - Verificare che il $framework_G$ permetta l'aggiornamento movie per i grafici di tipo *Map Chart_G* (TV2.1.3).

C.3.3 Test TV3

Lo sviluppatore vuole verificare che il $framework_G$ fornisca un sistema di log per il tracciamento di eventuali errori.

Allo sviluppatore è richiesto di:

- Scrivere volutamente del codice errato per provocare errori in fase di esecuzione (TV3.1);
- Verificare che sia effettivamente presente un sistema di log per la visualizzazione degli errori provocati (TV3.2).

C.3.4 Test TV4

Lo sviluppatore vuole verificare che sia possibile configurare vari parametri sulle pagine create.

Allo sviluppatore è richiesto di:

- Verificare che sia possibile assegnare il titolo alla pagina (TV4.1);
- Verificare che sia possibile configurare il massimo numero di grafici inseribili nella pagina (TV4.2);
- Verificare che sia possibile cambiare il numero massimo di grafici inseribili nella pagina (TV4.3);
- Verificare che sia possibile rimuovere un grafico esistente dalla pagina (TV4.4).

C.3.5 Test TV5

Lo sviluppatore vuole verificare che sia possibile creare un grafico e visualizzarlo.

Allo sviluppatore è richiesto di:

- Verificare che sia possibile creare un grafico. (TV5.1)
Allo sviluppatore è richiesto di:
 - Verificare che sia possibile creare un grafico di tipo *Bar Chart_G* (TV5.1.1);
 - Verificare che sia possibile creare un grafico di tipo *Line Chart_G* (TV5.1.2);
 - Verificare che sia possibile creare un grafico di tipo *Map Chart_G* (TV5.1.3);
 - Verificare che sia possibile creare un grafico di tipo *Table_G* (TV5.1.4).
- Verificare che i grafici siano visibili (TV5.2);
- Verificare che i grafici siano visibili da un browser (TV5.3).

C.3.6 Test TV6

Lo sviluppatore vuole verificare che sia disponibile l'indirizzo di una *dashboard_G* per la visualizzazione dei grafici attivi.

Allo sviluppatore è richiesto di:

- Accedere all'indirizzo della *dashboard_G* (TV6.1);
- Verificare che la *dashboard_G* visualizzi i grafici attivi sul *server_G* (TV6.2).

C.3.7 Test TV7

L'utente vuole verificare che sia possibile visualizzare grafici tramite l'applicazione *Android_G*.

All'utente è richiesto di:

- Verificare che sia possibile inserire un indirizzo *URL_G* nell'apposita form. (TV7.1)
All'utente è richiesto di:
 - Verificare che venga visualizzato un messaggio di errore nel caso in cui l'indirizzo non sia valido (TV7.1.1);

- Verificare che venga visualizzato un messaggio di errore nel caso in cui l'indirizzo URL_G inserito non corrisponda ad una istanza di Norris (TV7.1.2).
- Verificare che sia possibile selezionare e visualizzare il grafico tra quelli presenti nella pagina corrispondente all'indirizzo URL_G inserito (TV7.2);
- Verificare che l'applicazione preveda una lista di grafici recenti selezionabili e visualizzabili (TV7.3);
- Verificare che l'applicazione preveda una sezione "Impostazioni". (TV7.4)
All'utente è richiesto di:
 - Verificare che la sezione "Impostazioni" permetta di cambiare la lingua dell'applicazione, scegliendo tra quella italiana e quella inglese (TV7.4.1);
 - Verificare che la sezione "Impostazioni" permetta di visualizzare le informazioni riguardanti la versione dell'applicazione e il team di sviluppo (TV7.4.2).

C.3.8 Test TV8

Lo sviluppatore vuole verificare che sia possibile configurare vari parametri sui grafici in fase di creazione.

Allo sviluppatore è richiesto di:

- Verificare che sia possibile assegnare un titolo ad un grafico (TV8.1);
- Verificare che sia possibile inserire dei dati in un grafico (TV8.2);
- Verificare che sia possibile scegliere il formato di visualizzazione dei dati (TV8.3);
- Verificare che sia possibile scegliere i colori dei dati (TV8.4);
- Verificare che sia possibile impostare il colore per ogni cella di un grafico di tipo $Table_G$ (TV8.5);
- Verificare che sia possibile scegliere se visualizzare o meno la legenda di un grafico (TV8.6);
- Verificare che sia possibile scegliere dove posizionare la legenda di un grafico (TV8.7);
- Verificare che sia possibile assegnare un nome agli assi di un grafico (TV8.8);
- Verificare che sia possibile scegliere se visualizzare o meno la griglia degli assi (TV8.9);
- Verificare che sia possibile scegliere l'orientamento delle barre di un grafico di tipo $Bar\ Chart_G$ tra verticale e orizzontale (TV8.10);
- Verificare che sia possibile scegliere il punto centrale della mappa di un grafico di tipo $Map\ Chart_G$ (TV8.11);
- Verificare che sia possibile impostare le dimensioni dell'area di mappa visualizzata di un grafico di tipo $Map\ Chart_G$ (TV8.12);
- Verificare che sia possibile scegliere se visualizzare o meno i bordi di un grafico di tipo $Table_G$ (TV8.13);

- Verificare che sia possibile assegnare un header alle colonne di un grafico di tipo *Table_G* (TV8.14);
- Verificare che sia possibile ordinare con criterio alfanumerico, crescente o decrescente, gli elementi di un grafico di tipo *Table_G* rispetto ad una colonna (solo con aggiornamento in place) (TV8.15);
- Verificare che sia possibile scegliere se l'aggiunta di nuovi elementi ad un grafico di tipo *Table_G* avviene in testa o in coda (solo con aggiornamento stream) (TV8.16);
- Verificare che sia possibile specificare il massimo numero di elementi di un grafico di tipo *Table_G* visualizzati contemporaneamente (TV8.17).

C.3.9 Test TV9

Lo sviluppatore vuole verificare che sia possibile modificare dei parametri ad un grafico già creato.

Allo sviluppatore è richiesto di:

- Verificare che sia possibile aggiornare i dati di un grafico (TV9.1);
- Verificare che sia possibile modificare il colore per ogni cella di un grafico di tipo *Table_G* (TV9.2).

C.3.10 Test TV10

L'utente vuole verificare che sia possibile visualizzare una *dashboard_G* che monitora gli spostamenti degli autobus della società APS Holding nella città di Padova.

All'utente è richiesto di:

- Verificare che il grafico utilizzato per visualizzare la posizione degli autobus sia di tipo *Map Chart_G* (TV10.1);
- Selezionare una linea degli autobus (TV10.2);
- Verificare che nel grafico appaiano i dati degli autobus appartenenti alla linea selezionata (TV10.3).

C.4 Test di Unità

Test	Descrizione	Stato
TU1	Si verifica che i metodi get/set ritornino l'oggetto atteso	Successo
TU2	Si verifica che, dato un determinato input, il <i>routing_G</i> della richiesta venga risolto correttamente	Successo
TU3	Si verifica che la pagina sulla quale poi andranno inseriti i grafici venga creata correttamente	Successo
TU4	Si verifica che dato un <i>URL_G</i> corretto ad una istanza di Norris esistente vengano ritornati tutti i grafici presenti	Successo
TU5	Si verifica che un oggetto della classe grafico, dati determinati input, venga costruito nel modo corretto secondo quanto atteso	Successo
TU6	Si verifica che il metodo restituisca l'errore nel formato Error di Norris atteso	Successo
TU7	Si verifica che la pagina venga creata delle dimensioni che sono passate come input	Successo
TU9	Si verifica che il metodo, dati determinati input, aggiunga correttamente il grafico alla pagina	Successo
TU10	Si verifica che dato un certo input il numero di grafici per riga venga impostato correttamente	Successo
TU11	Si verifica che il metodo, dati determinati input, aggiorni il set di dati del grafico secondo la tipologia settata per il grafico	Successo
TU13	Si verifica che all'inserimento di un grafico nella pagina il vincolo di grafici per riga venga rispettato e l'eventuale errore sia gestito correttamente	Successo
TU14	Si verifica che il metodo di inserimento in testa ad un grafico di tipo <i>Table_G</i> inserisca effettivamente il dato nella posizione corretta	Successo
TU15	Si verifica che il metodo di inserimento in coda ad un grafico di tipo <i>Table_G</i> inserisca effettivamente il dato nella posizione corretta	Successo

Test	Descrizione	Stato
TU16	Si verifica che il metodo, dato un determinato input, imposti correttamente lo sfondo della cella	Successo
TU20	Si verifica che il metodo setti correttamente il campo che rappresenta il numero di grafici per riga	Successo
TU27	Si verifica che l'oggetto Pagina venga creato correttamente	Successo
TU29	Si verifica che il metodo restituisca l'errore in formato <i>JSON_G</i> atteso	Successo
TU30	Si verifica che il metodo notifichi correttamente ai <i>client_G</i> la modifica di un dato	Successo
TU32	Si verifica che il metodo ProgressiveID, legga e restituisca correttamente il valore dell'Id progressivo da assegnare e poi vada ad incrementarlo	Successo
TU37	Si verifica che l'ordine di visualizzazione dei grafici coincida con l'ordine in cui sono stati inseriti	Successo
TU39	Si verifica che il metodo ritorni un array contenente un set di colori e che il formato sia conforme a quanto atteso	Successo
TU40	Si verifica che il gestore delle connessioni tra <i>client_G</i> e <i>server_G</i> venga correttamente avviato ed eventuali errori vengano opportunamente gestiti	Successo
TU41	Si verifica che il metodo salvi correttamente il riferimento al grafico/pagina quando questa viene creata	Successo
TU42	Si verifica che, dato un determinato input, il metodo ritorni una pagina o un grafico e questo sia quanto atteso	Successo
TU43	Si verifica che il metodo, dati determinati input, costruisca correttamente un oggetto della classe Norris	Successo

Test	Descrizione	Stato
TU44	Si verifica che il metodo, dati determinati input, costruisca correttamente un oggetto della classe PageRouter	Successo
TU45	Si verifica che nel caso in cui alcune opzioni passate per la creazione di una pagina siano incomplete o mancati il metodo restituisca un oggetto in cui le eventuali mancanze sono state sostituite da valori di default	Successo
TU46	Si verifica che il metodo restituisca correttamente i riferimenti agli array delle risorse attive nell'istanza di Norris	Successo
TU47	Si verifica che il metodo salvi correttamente un riferimento al grafico quando questo viene creato	Successo
TU48	Si verifica che il metodo, dato in input il valore per l'orientamento delle barre, lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU49	Si verifica che il metodo, dato in input il valore per settare la posizione della legenda nei grafici di tipo <i>Bar Chart_G</i> e <i>Line Chart_G</i> , lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU50	Si verifica che il metodo, dato in input il valore per settare la posizione della legenda nei grafici di tipo <i>Map Chart_G</i> , lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU51	Si verifica che il metodo, dato in input il valore per settare lo zoom nei grafici di tipo <i>Map Chart_G</i> , lanci l'errore corrispondente nel caso questo non sia compreso nel range possibile per la mappa	Successo
TU52	Si verifica che il metodo, dato in input il valore per settare il pathMode nei grafici di tipo <i>Map Chart_G</i> , lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo

Test	Descrizione	Stato
TU53	Si verifica che il metodo, dato in input il valore per settare l'ordinamento dei dati nei grafici di tipo <i>TableG</i> , lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU54	Si verifica che il metodo, dato in input il valore per settare la posizione di inserimento dei nuovi dati, lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU55	Si verifica che il metodo, dato in input il valore per settare il numero di righe visualizzabili per scheda in un grafico di tipo <i>TableG</i> , lanci l'errore corrispondente nel caso questo non sia maggiore di zero	Successo
TU56	Si verifica che il metodo, dato in input un array di colori, lanci l'errore corrispondente nel caso questi non siano nel formato esadecimale	Successo
TU57	Si verifica che il metodo, dato in input un oggetto colore, lanci l'errore corrispondente nel caso questo non sia nel formato esadecimale	Successo
TU58	Si verifica che il metodo, dati in input i valori per settare il colore di sfondo delle celle di una riga o colonna e il colore del font, lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU59	Si verifica che il metodo, dati in input i valori per settare il colore della cella e del carattere, lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU60	Si verifica che il metodo, dato in input un valore per settare la proprietà "series", lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU61	Si verifica che il metodo, dato in input un valore per settare la proprietà "valueType", lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo

Test	Descrizione	Stato
TU62	Si verifica che il metodo, dato in input un valore per settare la proprietà "decimals", lanci l'errore corrispondente nel caso questo non sia un numero o non sia compreso tra 0 e 6	Successo
TU63	Si verifica che il metodo, dato in input un valore per settare il formato dei dati di un grafico di tipo <i>Table_G</i> , lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU64	Si verifica che il metodo, dato in input un valore per settare la proprietà "LabelsLimit" e "rowsLimit", lanci l'errore corrispondente nel caso questo non sia tra i valori accettati	Successo
TU65	Si verifica che il metodo, dato un determinato input, converta correttamente i dati grezzi che arrivano dal <i>back-end_G</i> in dati per il modello	Successo
TU66	Si verifica che il metodo aggiorni correttamente i dati con la modalità inPlace	Successo
TU67	Si verifica che il metodo aggiorni correttamente i dati con la modalità stream e che il limite massimo di valori non venga superato	Successo
TU68	Si verifica che la connessione al <i>namespace_G</i> specificato venga aperta correttamente	Successo
TU69	Si verifica che la callback che resta in ascolto di un determinato evento venga attivata correttamente	Successo
TU70	Si verifica che il metodo divida correttamente i dati grezzi che riceve in righe e colonne	Successo
TU71	Si verifica che il metodo converta l'array in un oggetto LatLng correttamente e sia poi possibile usarlo nelle <i>Google Maps API_G</i>	Successo
TU72	Si verifica che il metodo crei correttamente la linea colorata sulla mappa	Successo
TU73	Si verifica che il metodo, date le coordinate di un punto, crei il segnalino sulla mappa	Successo

Test	Descrizione	Stato
TU74	Si verifica che il metodo, a partire da 2 coordinate generi un percorso stradale sulla mappa	Successo
TU75	Si verifica che il metodo, dato un determinato input, posizioni correttamente la legenda dentro la mappa	Successo
TU76	Si verifica che il metodo legga correttamente l' <i>URL_G</i> ed effettui correttamente la chiamata al <i>back-end_G</i>	Successo
TU77	Si verifica che il metodo, dati determinati input, aggiorni correttamente i punti presenti sulla mappa	Successo

Tabella 5: Test di Unità

C.4.1 Tracciamento Test di Unità - Metodi

Test	Metodi
TU1	Norris::Lib::PresentationLayer::Page::getPageInfo() Norris::Lib::BusinessLayer::BarChartController::getChartInfo() Norris::Lib::BusinessLayer::LineChartController::getChartInfo() Norris::Lib::BusinessLayer::MapChartController::getChartInfo() Norris::Lib::BusinessLayer::SocketController::setSocket() Norris::Lib::BusinessLayer::SocketController::socketNamespace() Norris::Lib::BusinessLayer::TableController::getChartInfo() Norris::Lib::Utils::SocketService::setSocket() Norris::Lib::Utils::SocketService::getSocketNamespace() Norris::Lib::PresentationLayer::BarChart::getChartInfo() Norris::Lib::PresentationLayer::LineChart::getChartInfo() Norris::Lib::PresentationLayer::Table::getChartInfo() Norris::Lib::BusinessLayer::PageController::getPageInfo()
TU2	Norris::Lib::PresentationLayer::PageRouter::PageRouter()
TU3	Norris::Lib::BusinessLayer::PageController::createPage()
TU4	Norris::Lib::PresentationLayer::PageRouter::PageRouter()

Test	Metodi
TU5	Norris::Lib::BusinessLayer::BarChartController::createBarChart() Norris::Lib::BusinessLayer::LineChartController::createLineChart() Norris::Lib::BusinessLayer::MapChartController::createMapChart() Norris::Lib::BusinessLayer::TableController::createTable()
TU6	Norris::Lib::Utils::NorrisError::NorrisError()
TU7	Norris::Lib::BusinessLayer::PageController::createPage()
TU9	Norris::Lib::PresentationLayer::Page::addGraph() Norris::Lib::BusinessLayer::PageController::addGraphToPage()
TU10	Norris::Lib::BusinessLayer::PageController::createPage()
TU11	Norris::Lib::BusinessLayer::BarChartController::updateInPlace() Norris::Lib::BusinessLayer::LineChartController::updateStream() Norris::Lib::BusinessLayer::MapChartController::updateMovie() Norris::Lib::BusinessLayer::TableController::updateStream() Norris::Lib::PresentationLayer::BarChart::updateInPlace() Norris::Lib::PresentationLayer::LineChart::updateInPlace() Norris::Lib::PresentationLayer::LineChart::updateStream() Norris::Lib::PresentationLayer::Table::updateInPlace() Norris::Lib::PresentationLayer::Table::updateStream() Norris::Lib::BusinessLayer::LineChartController::updateInPlace() Norris::Lib::BusinessLayer::TableController::updateInPlace()
TU13	Norris::Lib::BusinessLayer::PageController::addGraphToPage()
TU14	Norris::Lib::BusinessLayer::TableController::updateStream()
TU15	Norris::Lib::PresentationLayer::Table::updateStream()
TU16	Norris::Lib::BusinessLayer::TableController::createTable()
TU20	Norris::Lib::BusinessLayer::PageController::createPage()
TU27	Norris::Lib::BusinessLayer::PageController::createPage()
TU29	Norris::Lib::Utils::NorrisError::NorrisError()
TU30	Norris::Lib::BusinessLayer::SocketController::sendUpdate() Norris::Lib::Utils::SocketService::sendUpdate()
TU32	Norris::Lib::Utils::ProgressiveID::ProgressiveID()
TU37	Norris::Lib::BusinessLayer::PageController::addGraphToPage()
TU39	Norris::Lib::Utils::ColorManager::ColorGenerator()

Test	Metodi
TU40	Norris::Lib::Utils::SocketService::connectionManager()
TU41	Norris::Lib::BusinessLayer::ActiveResourcesController::storePage()
TU42	Norris::Lib::BusinessLayer::ActiveResourcesController::retrieveGraph() Norris::Lib::BusinessLayer::ActiveResourcesController::retrievePage()
TU43	Norris::Lib::PresentationLayer::Norris::Norris()
TU44	Norris::Lib::PresentationLayer::PageRouter::PageRouter()
TU45	Norris::Lib::BusinessLayer::PageController::createPageOptions()
TU46	Norris::Lib::DataLayer::ActiveResources::ActiveResources()
TU47	Norris::Lib::BusinessLayer::ActiveResourcesController::storeGraph()
TU48	Norris::Lib::BusinessLayer::DataConsistency::checkOrientation()
TU49	Norris::Lib::BusinessLayer::DataConsistency::checkLegendPosition()
TU50	Norris::Lib::BusinessLayer::DataConsistency::checkMapLegendPosition()
TU51	Norris::Lib::BusinessLayer::DataConsistency::checkMapZoom()
TU52	Norris::Lib::BusinessLayer::DataConsistency::pathMode()
TU53	Norris::Lib::BusinessLayer::DataConsistency::checkOrderBy()
TU54	Norris::Lib::BusinessLayer::DataConsistency::checkInsertPosition()
TU55	Norris::Lib::BusinessLayer::DataConsistency::checkDisplayedLines()
TU56	Norris::Lib::BusinessLayer::DataConsistency::checkColors()
TU57	Norris::Lib::BusinessLayer::DataConsistency::checkHex()
TU58	Norris::Lib::BusinessLayer::DataConsistency::checkColorArray()
TU59	Norris::Lib::BusinessLayer::DataConsistency::checkColorMatrix()
TU60	Norris::Lib::BusinessLayer::DataConsistency::checkSeries()
TU61	Norris::Lib::BusinessLayer::DataConsistency::checkValueType()
TU62	Norris::Lib::BusinessLayer::DataConsistency::checkDecimals()
TU63	Norris::Lib::BusinessLayer::DataConsistency::checkTableFormat()
TU64	Norris::Lib::BusinessLayer::DataConsistency::checkBounds()

Test	Metodi
TU65	Norris::NorrisApp::Services::TableSvc::fillData()
TU66	Norris::NorrisApp::Services::TableSvc::inPlaceUpd()
TU67	Norris::NorrisApp::Services::TableSvc::streamUpd()
TU68	Norris::NorrisApp::Services::SocketsSvc::open()
TU69	Norris::NorrisApp::Services::SocketsSvc::on()
TU70	Norris::NorrisApp::Services::FrontSvc::createRows()
TU71	Norris::NorrisApp::Services::MapSvc::cnvLatLong()
TU72	Norris::NorrisApp::Services::MapSvc::createPolyline()
TU73	Norris::NorrisApp::Services::MapSvc::createMarker()
TU74	Norris::NorrisApp::Services::MapSvc::buildPath()
TU75	Norris::NorrisApp::Services::MapSvc::buildLegend()
TU76	Norris::NorrisApp::Services::FirstConnectSvc::FirstConnectSvc()
TU77	Norris::NorrisApp::Services::MapSvc::updateMovie()

Tabella 6: Tracciamento Test di Unità - metodi

D Risultati della verifica

D.1 Gestione della revisione

D.1.1 Gestione anomalie riscontrate

Se durante la fase di verifica i *Verificatori* dovessero incontrare una o più anomalie, dovranno provvedere ad aprire un *ticket_G* usando il sistema *Redmine_G*. Per maggiori dettagli consultare la sezione dedicata del documento *NormeDiProgetto_ver3.0.0*.

L'identificazione delle anomalie ne permette la correzione e informa il *Responsabile* di progetto sullo stato di avanzamento del prodotto. Il gruppo ha scelto di distinguere e catalogare le anomalie in maniera tale da differenziare la tipologia di correzioni da apportare e con quale priorità questa debba essere fatta. Di seguito vengono elencate le definizioni di anomalie adottate dal gruppo:

- **Error:** differenza riscontrata tra il risultato di una computazione e il valore teorico atteso (sforamento dei range fissati);

- **Fault:** un passo, un processo o un dato definito in modo erraneo, corrisponde a quanto viene definito come *bug_G* (non attuazione di norme tipografiche);
- **Failure:** il risultato di un fault (discostamento funzionalità con quanto emerso dai requisiti);
- **Mistake:** errore umano.

D.1.2 Controlli per la qualità di processo

Le procedure di controllo per la qualità di processo sono finalizzate a migliorare la qualità del prodotto e/o diminuire i costi e tempi di sviluppo. Esistono due approcci principali:

- **A maturità di processo:** riflette le buone pratiche di management e tecniche di sviluppo. L'obiettivo primario è la qualità del prodotto e la prevedibilità dei processi;
- **Agile:** sviluppo iterativo senza l'overhead della documentazione e di tutti gli aspetti predeterminabili. Ha come caratteristica la responsività ai cambiamenti dei requisiti cliente e uno sviluppo rapido.

Il gruppo ha deciso di utilizzare l'approccio "A maturità di processo", in quanto più adatto ad un team inesperto. Con una visione proattiva si cerca di avere maggior controllo e previsione sulle attività da svolgere. Questa viene anche indicata come buona pratica per gruppi poco esperti.

Il processo con maggiore influenza sulla qualità del sistema non è quello di sviluppo ma quello di progettazione. È qui che le capacità e le esperienze dei singoli danno un contributo decisivo.

Un'anomalia corrisponde a:

- Mancato rispetto delle norme tipografiche;
- 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.

D.2 Revisione dei Requisiti

Nel periodo precedente la consegna della Revisione dei Requisiti, comprendente la fase **A**, sono stati sottoposti a verifica tutti i processi ed i documenti realizzati secondo quanto descritto nel documento *NormeDiProgetto_ver3.0.0*. In particolare si è ricorsi all'utilizzo della tecnica walkthrough per tutti i documenti data la scarsa esperienza dei membri del gruppo.

Questa attività di analisi statica ha permesso di rilevare diversi errori che sono stati prontamente segnalati e gestiti in maniera conforme a quanto descritto nel documento *NormeDiProgetto_ver3.0.0*. Per quanto riguarda gli errori più frequentemente rilevati si è inoltre provveduto ad aggiungerli nell'apposita lista di controllo.

Sono stati effettuati inoltre i tracciamenti necessari tramite l'applicazione *FIRE_G*.

Grazie al continuo controllo sui processi attuati, sulla base di quanto descritto nel

documento *NormeDiProgetto_ver3.0.0*, è stato possibile monitorare costantemente l'avanzamento delle attività ed avere un indicatore per quanto riguarda l'utilizzo di risorse rispetto a quanto pianificato nel documento *PianoDiProgetto_ver4.0.0*.

Riguardo l'utilizzo di risorse, vengono riportati i valori riguardanti rispettivamente lo *Schedule Variance* (espresso in ore) ed il *Budget Variance*, suddivisi per documento.

Documento	SV	BV
<i>NormeDiProgetto_ver1.4.0</i>	-1	-20€
<i>PianoDiQualifica_ver1.4.0</i>	0	0€
<i>AnalisiRequisiti_ver1.4.0</i>	-2	-40€
<i>StudioDiFattibilità_ver1.4.0</i>	–	–
<i>PianoDiProgetto_ver1.4.0</i>	3	90€
<i>Glossario_ver1.4.0</i>	0	0€

Tabella 7: Risultati indice di *BV* e *SV* - Revisione dei Requisiti

Complessivamente in questo periodo sono stati registrati i seguenti valori:

- **SV** = 0 ore
- **BV** = 30€

Dai valori riscontrati è possibile dedurre che:

- Grazie ai periodi di *slack_G* che sono stati inseriti durante il periodo di pianificazione tutte le attività sono state svolte rispettando i tempi pianificati come indicato nel documento *PianoDiProgetto_ver4.0.0*;
- Il valore del *BV_G* risulta positivo in quanto le attività hanno richiesto una quantità complessiva di risorse minore rispetto a quanto pianificato.

Per finire, tutti i documenti sono stati sottoposti allo script per 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 8: Risultati indice di *Gulpease_G* - Revisione dei Requisiti

Come indicato dai risultati presentati nella precedente tabella, tutti i valori relativi ai documenti rientrano nel range ottimale che il gruppo aveva prefissato.

D.2.1 Dettaglio esito Revisione dei Requisiti

Per la Revisione dei Requisiti il Committente ha segnalato alcune correzioni da apportare ([lista correzioni](#)); di seguito sono descritte le modifiche che il gruppo ha appor-

tato ai vari documenti per migliorarne l'organizzazione ed i contenuti secondo quanto segnalato:

- **Norme di Progetto:** i contenuti del documento sono stati riorganizzati per processi, attività, procedure e strumenti. Inoltre, è stato rivisto lo stile di presentazione aggiungendo immagini e diagrammi per spiegarne meglio il contenuto. Per concludere, è stato inserito il dettaglio sulle regole e procedure per la rotazione dei ruoli;
- **Analisi dei Requisiti:** sono state apportate le modifiche suggerite ai *casi d'uso_G* e requisiti indicati, sono stati specificati maggiormente dove richiesto e sono state modificate le dimensioni delle immagini per renderne leggibile il contenuto. I *casi d'uso_G* e i requisiti riguardanti l'applicazione *Android_G* e l'applicazione di esempio, invece, sono stati inseriti nel documento durante la fase **B**, quindi in data successiva alla consegna del materiale per la Revisione dei Requisiti ma precedente all'applicazione delle correzioni segnalate;
- **Piano di Qualifica:** gli obiettivi di qualità sono stati espressi in maniera quantitativa e direttamente associati a specifiche metriche e misure. Inoltre, alcuni contenuti tra loro correlati sono stati riorganizzati per rappresentare meglio il legame tra gli stessi;
- **Piano di progetto:** è stato rivisto l'utilizzo della terminologia come segnalato, l'analisi dei rischi è stata migliorata aggiungendo una sezione incrementale riguardante il riscontro effettivo dei rischi durante le varie fasi, mentre la pianificazione è rimasta invariata in quanto la presenza dei rischi costituisce un fattore già considerato al momento della stesura del documento. È stata rimossa la sezione "consuntivo a finire" ed è stata sostituita con una ulteriore appendice contenente il consuntivo parziale delle prime due fasi non rendicontate, in modo da rendere manifesto l'investimento effettuato.

D.3 Revisione di Progettazione

Nel periodo precedente la consegna della Revisione di Progettazione, comprendente le fasi **B** e **C**, sono stati sottoposti a verifica tutti i processi ed i documenti realizzati secondo quanto descritto nel documento *NormeDiProgetto_ver3.0.0*. In particolare si è ricorsi all'utilizzo della tecnica walkthrough per il documento *SpecificaTecnica_ver2.0.0* in quanto non presente nella precedente fase. Per quanto riguarda gli incrementi ai documenti già esistenti nella precedente fase invece è stata utilizzata la tecnica inspection con particolare attenzione agli errori indicati nella lista di controllo.

Queste attività di analisi statica hanno permesso di rilevare diversi errori che sono stati prontamente segnalati e gestiti in maniera conforme a quanto descritto nel documento *NormeDiProgetto_ver3.0.0*. Per quanto riguarda gli errori più frequentemente rilevati si è inoltre provveduto ad aggiungerli nell'apposita lista di controllo.

Sono stati effettuati inoltre i tracciamenti necessari tramite l'applicazione *FIRE_G*.

Grazie al continuo controllo sui processi attuati, sulla base di quanto descritto nel documento *NormeDiProgetto_ver3.0.0*, è stato possibile monitorare costantemente l'avanzamento delle attività ed avere un indicatore per quanto riguarda l'utilizzo di risorse rispetto a quanto pianificato nel documento *PianoDiProgetto_ver4.0.0*.

Riguardo l'utilizzo di risorse, vengono riportati i valori riguardanti rispettivamente lo *Schedule Variance* (espresso in ore) ed il *Budget Variance*, suddivisi per documento.

Documento	SV	BV
<i>NormeDiProgetto_ver2.0.0</i>	3	60€
<i>PianoDiQualifica_ver2.0.0</i>	-3	-45€
<i>AnalisiRequisiti_ver3.0.0</i>	5	125€
<i>PianoDiProgetto_ver3.0.0</i>	5	140€
<i>Glossario_ver3.0.0</i>	0	0€
<i>SpecificaTecnica_ver1.0.0</i>	-5	-103€

Tabella 9: Risultati indice di *BV* e *SV* - Revisione di Progettazione

Complessivamente in questo periodo sono stati registrati i seguenti valori:

- **SV** = 5 ore
- **BV** = 177€

Dai valori riscontrati è possibile dedurre che:

- Grazie ai periodi di *slack_G* che sono stati inseriti durante il periodo di pianificazione tutte le attività sono state svolte rispettando i tempi pianificati come indicato nel documento *PianoDiProgetto_ver4.0.0*;
- Il valore del *BV_G* risulta positivo in quanto le attività hanno richiesto una quantità complessiva di risorse minore rispetto a quanto pianificato.

Per finire, tutti i documenti sono stati sottoposti allo script per il calcolo dell'indice di leggibilità di *Gulpease_G*. I risultati ottenuti sono:

Documento	Valore	Esito
<i>NormeDiProgetto_ver2.0.0</i>	56,27	superato
<i>PianoDiQualifica_ver2.0.0</i>	54,77	superato
<i>AnalisiRequisiti_ver3.0.0</i>	81,19	superato
<i>SpecificaTecnica_ver1.0.0</i>	65,07	superato
<i>PianoDiProgetto_ver3.0.0</i>	57,39	superato
<i>Glossario_ver3.0.0</i>	50,03	superato

Tabella 10: Risultati indice di *Gulpease_G* - Revisione di Progettazione

Come indicato dai risultati presentati nella precedente tabella, tutti i valori relativi ai documenti rientrano nel range ottimale che il gruppo aveva prefissato.

D.3.1 Dettaglio esito Revisione di Progettazione

Per la Revisione di Progettazione il Committente ha segnalato alcune correzioni da apportare ([lista correzioni](#)); di seguito sono descritte le modifiche che il gruppo ha apportato ai vari documenti per migliorarne l'organizzazione ed i contenuti secondo quanto segnalato:

- **Correzioni generali:** a seguito della segnalazione di un dettaglio non sufficiente sui diari delle modifiche si è provveduto per le nuove modifiche a specificare più nel dettaglio quali sono state le modifiche apportate al documento includendo anche in quale sezione sono state spostate alcune sezioni a seguito delle osservazioni ricevuto. Anche per quanto riguarda i verbali sono state apportate delle modifiche in maniera tale che ora le decisioni prese siano tracciabili.
- **Norme di Progetto:** i contenuti del documento, prevalentemente per quanto riguarda il capitolo \$4 sono stati riorganizzati per processi, attività, procedure e strumenti. Si è certato di prestare attenzione nel mettere su piani differenti processi, strumenti e attività;
- **Analisi dei Requisiti:** migliorato il contenuto riguardante il flusso principale degli eventi per ogni caso d'uso;
- **Piano di Qualifica:** rimosse alcune sezioni dal documento e trasferite nelle *NormeDiProgetto_ver3.0.0*. È stato rimosso il capitolo \$3, riguardante il la gestione della revisione e si è provveduto ad integrarlo e migliorarlo nell'Appendice D, inoltre è stata rivista la definizione del *PDCA_G*;
- **Specifica Tecnica:** a seguito delle segnalazioni ricevute e ad alcuni colloqui avuti con il *Proponente_G* il documento ha subito importanti modifiche per quanto ne riguarda i contenuti. Si è comunque provveduto a rivedere i diagrammi di attività sulla base di quanto segnalato nell'esito della revisione.

D.4 Revisione di Qualifica

Nel periodo precedente la consegna della Revisione di Qualifica, comprendente la fase **C**, tutti i processi ed i documenti sono stati sottoposti a verifica seguendo quanto riportato nel documento *NormeDiProgetto_ver3.0.0*. Per quanto riguarda i nuovi documenti presentati, *ManualeUtente_ver1.0.0* e *DefinizioneDiProdotto_ver1.0.0*, si è ricorsi alla tecnica del walkthrough. Per quanto riguarda invece i documenti precedentemente creati è stata usata la tecnica dell'inspection ponendo particolare attenzione agli errori già specificati nella lista di controllo. Tutti gli errori trovati sono stati segnalati seguendo le modalità descritte nelle *NormeDiProgetto_ver3.0.0* e provvedendo, qual ora fossero frequenti ad aggiornare la lista di controllo. Come per la revisione precedente il tracciamento è stato effettuato utilizzando il sistema di tracking da noi realizzato, il quale ha subito dei miglioramenti per far fronte alle nuove esigenze che sono emerse. Il gruppo ha continuato a monitorare tutti i processi attuati con lo scopo di migliorarne l'efficienza ed avere un continuo indicatore sull'utilizzo delle risorse.

Per questa fase i valori rilevati, per quanto riguarda *Schedule Variance* e *Budget Variance* sono i seguenti:

Documento	SV	BV
<i>NormeDiProgetto_ver3.0.0</i>	0	0€
<i>PianoDiQualifica_ver3.0.0</i>	1	30€
<i>AnalisiRequisiti_ver4.0.0</i>	4	75€
<i>PianoDiProgetto_ver4.0.0</i>	2	45€
<i>Glossario_ver4.0.0</i>	0	0€
<i>SpecificaTecnica_ver2.0.0</i>	-5	-103€
<i>ManualeUtente_ver1.0.0</i>	1	15€
<i>DefinizioneDiProdotto_ver1.0.0</i>	-2	-50€

Tabella 11: Risultati indice di *BV* e *SV* - Revisione di Qualifica

Il totale complessivo per questo periodo è stato di:

- **SV** = 1 ora
- **BV** = 12€

Oltre al controllo sull'utilizzo delle risorse si è ricorsi anche all'analisi dei documenti utilizzando l'*Indice di Gulpease*. I risultati emersi sono stati i seguenti:

Documento	Valore	Esito
<i>NormeDiProgetto_ver3.0.0</i>	56.57	superato
<i>PianoDiQualifica_ver3.0.0</i>	56.31	superato
<i>AnalisiRequisiti_ver4.0.0</i>	81.01	superato
<i>PianoDiProgetto_ver4.0.0</i>	56.21	superato
<i>Glossario_ver4.0.0</i>	50.12	superato
<i>SpecificaTecnica_ver2.0.0</i>	63.25	superato
<i>ManualeUtente_ver1.0.0</i>	58.00	superato
<i>DefinizioneDiProdotto_ver1.0.0</i>	70.06	superato

Tabella 12: Risultati indice di *BV* e *SV* - Revisione di Qualifica

D.4.1 Misurazioni sul codice

Per questa revisione è stato presentato il codice del framework in fase di realizzazione. Come per i documenti anche il codice è stato sottoposto a verifica; per avere una descrizione delle metriche verificate è possibile consultare la sezione 2.7.1. Per fornire miglior dettaglio e chiarezza i valori rilevati sono stati separati tra *Back-end_G* e *Front-end*, nel caso in cui la metrica non dovesse rientrare almeno nel range di accettazione verrà fornita anche la motivazione per tale incongruenza.

Vengono ora riportate le misure rilevate attuando sia analisi statica che dinamica:

Complessità ciclomatica

- *Back-end_G*
 - Media: 3.2
 - Massima: 15
- *Front-end_G*
 - Media: 2.6
 - Massima: 11

Tutti i moduli rientrano nel range di accettazione.

Parametri per metodo

- *Back-end_G*
 - Media: 3.8
 - Massima: 7
- *Front-end_G*
 - Media: 4.2
 - Massima: 7

Tutti i moduli rientrano nel range di accettazione.

Rapporto codice/commenti

- *Back-end_G*
 - Media: 0.42
 - Minima: 0.27
- *Front-end_G*
 - Media: 0.35
 - Minima: 0.26

Tutti i moduli rientrano nel range di accettazione.

Livello di annidamento

- *Back-end_G*
 - Media: 4.1
 - Massima: 6
- *Front-end_G*
 - Media: 2.5
 - Massima: 4

Tutti i moduli rientrano nel range di accettazione.

A causa di strumenti insoddisfacenti, la metrica relativa al numero di chiamate innestate non può essere calcolata. Il gruppo ha comunque provveduto a migliorare gli strumenti utilizzati per aumentare il livello di automazione del processo di calcolo delle metriche.