

# NORRIS Framework



FLAMETECH Inc.

## Piano di Qualifica

### Informazioni sul documento

|                        |  |
|------------------------|--|
| Versione               | 2.0.0  |
| Redazione              | Meneguzzo Francesco<br>Sartor Michele                                    |
| Verifica               | Faggin Andrea  |
| Responsabile           | Zanetti Davide   |
| Uso                    | Esterno  |
| Lista di distribuzione | <b>FlameTech Inc.</b><br>Prof. Vardanega Tullio<br>Prof. Cardin Riccardo |

### Descrizione

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

| <b>Stato</b>   | <b>Modifica</b>   | <b>Autore</b>          | <b>Ruolo</b> | <b>Data</b> | <b>Versione</b> |
|----------------|---|------------------------|--------------|-------------|-----------------|
| Approvato      | Approvazione documento  | Zanetti<br>Davide      | Responsabile | 2015/04/10  | 2.0.0           |
| Verificato     | Verifica documento  | Faggin<br>Andrea       | Verificatore | 2015/04/10  | 1.5.0           |
| In Lavorazione | Stesura appendice D.2   | Sartor<br>Michele      | Verificatore | 2015/04/09  | 1.4.4           |
| In Lavorazione | Stesura appendice D.1.1                                       | Meneguzzo<br>Francesco | Verificatore | 2015/03/10  | 1.4.3           |
| In Lavorazione | Stesura appendice<br>Pianificazione dei Test                  | Sartor<br>Michele      | Amminis.     | 2015/03/02  | 1.4.2           |
| In Lavorazione | Applicazione Correzioni<br>RR - riorganizzazione<br>contenuti | Sartor<br>Michele      | Amminis.     | 2015/02/27  | 1.4.1           |
| Approvato      | Approvazione documento  | Sartor<br>Michele      | Responsabile | 2015/01/15  | 1.4.0           |
| Verificato     | Verifica documento  | Cardin<br>Andrea       | Verificatore | 2015/01/08  | 1.3.0           |
| In Lavorazione | Stesura resoconto della<br>verifica                           | Meneguzzo<br>Francesco | Verificatore | 2015/01/07  | 1.2.0           |
| Verificato     | Verifica documento  | Meneguzzo<br>Francesco | Verificatore | 2014/12/10  | 1.1.0           |
| In Lavorazione | Stesura sezione sulle<br>metriche                             | Zanetti<br>Davide      | Amminis.     | 2014/12/07  | 1.0.3           |
| In Lavorazione | Stesura sezione gestione<br>della revisione                   | Zanetti<br>Davide      | Amminis.     | 2014/12/05  | 1.0.2           |
| In Lavorazione | Stesura sezione obiettivi<br>di qualità                       | Persegona<br>Mattia    | Analista     | 2014/12/03  | 1.0.1           |
| In Lavorazione | Iniziata stesura<br>documento                                 | Zanetti<br>Davide      | Amminis.     | 2014/12/02  | 1.0.0           |

## Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduzione</b>                                 | <b>1</b>  |
| 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         |
| <b>2</b> | <b>Obiettivi di qualità</b>                         | <b>3</b>  |
| 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      | Tecniche di Analisi . . . . .                       | 5         |
| 2.7.1    | Analisi Statica . . . . .                           | 5         |
| 2.7.1.1  | Inspection . . . . .                                | 5         |
| 2.7.1.2  | Walkthrough . . . . .                               | 6         |
| 2.7.2    | Analisi dinamica . . . . .                          | 6         |
| 2.7.2.1  | Test di unità . . . . .                             | 6         |
| 2.7.2.2  | Test di integrazione . . . . .                      | 7         |
| 2.7.2.3  | Test di regressione . . . . .                       | 7         |
| 2.7.2.4  | Test di sistema . . . . .                           | 7         |
| 2.7.2.5  | Test di accettazione . . . . .                      | 7         |
| 2.8      | Misure e Metriche adottate . . . . .                | 7         |
| 2.8.1    | Metriche per il software . . . . .                  | 8         |
| 2.8.1.1  | Livelli di annidamento . . . . .                    | 8         |
| 2.8.1.2  | Numero di parametri per metodo . . . . .            | 8         |
| 2.8.1.3  | Complessità ciclomatica . . . . .                   | 8         |
| 2.8.1.4  | Linee per metodo . . . . .                          | 8         |
| 2.8.1.5  | Rapporto tra commenti e codice . . . . .            | 9         |
| 2.8.1.6  | Chiamate innestate di metodi . . . . .              | 9         |
| 2.8.1.7  | Validazione <i>HTML<sub>G</sub></i> . . . . .       | 9         |
| 2.8.1.8  | Copertura del codice . . . . .                      | 9         |
| 2.8.1.9  | Numero campi per classe . . . . .                   | 9         |
| 2.8.2    | Metriche per i processi . . . . .                   | 9         |
| 2.8.2.1  | <i>Budget Variance<sub>G</sub></i> (BV) . . . . .   | 10        |
| 2.8.2.2  | <i>Schedule Variance<sub>G</sub></i> (SV) . . . . . | 10        |
| 2.8.3    | Metriche per i documenti . . . . .                  | 10        |
| 2.8.3.1  | Indice di <i>Gulpease<sub>G</sub></i> . . . . .     | 10        |
| <b>3</b> | <b>Gestione della revisione</b>                     | <b>12</b> |
| 3.1      | Gestione anomalie riscontrate . . . . .             | 12        |



|          |  |           |
|----------|--|-----------|
| <b>A</b> | <b>Qualità di Processo</b>                               | <b>13</b> |
| A.1      | Standard <i>ISO/IEC 15504<sub>G</sub></i> . . . . .      | 13        |
| A.2      | <i>Ciclo di Deming<sub>G</sub></i> . . . . .             | 14        |
| <b>B</b> | <b>Qualità di Prodotto</b>                               | <b>16</b> |
| B.1      | Standard <i>ISO/IEC 9126<sub>G</sub></i> . . . . .       | 16        |
| <b>C</b> | <b>Pianificazione dei Test</b>                           | <b>19</b> |
| C.1      | Test di integrazione . . . . .                           | 20        |
| C.1.1    | Descrizione dei test di integrazione . . . . .           | 20        |
| C.1.2    | Tracciamento componenti - test di integrazione . . . . . | 22        |
| C.2      | Test di sistema . . . . .                                | 24        |
| C.2.1    | Descrizione dei test di sistema . . . . .                | 24        |
| C.3      | Test di validazione . . . . .                            | 27        |
| C.3.1    | Test TV1 . . . . .                                       | 27        |
| C.3.2    | Test TV2 . . . . .                                       | 27        |
| C.3.3    | Test TV3 . . . . .                                       | 27        |
| C.3.4    | Test TV4 . . . . .                                       | 28        |
| C.3.5    | Test TV5 . . . . .                                       | 28        |
| C.3.6    | Test TV6 . . . . .                                       | 28        |
| C.3.7    | Test TV7 . . . . .                                       | 28        |
| C.3.8    | Test TV8 . . . . .                                       | 29        |
| C.3.9    | Test TV9 . . . . .                                       | 30        |
| C.3.10   | Test TV10 . . . . .                                      | 31        |
| <b>D</b> | <b>Risultati della verifica</b>                          | <b>32</b> |
| D.1      | Revisione dei Requisiti . . . . .                        | 32        |
| D.1.1    | Dettaglio esito Revisione dei Requisiti . . . . .        | 33        |
| D.2      | Revisione di Progettazione . . . . .                     | 34        |



## Elenco delle tabelle

|   |   |    |
|---|---|----|
| 2 | Tabella test di integrazione . . . . .                                  | 22 |
| 3 | Tabella Tracciamento Componenti - Test di integrazione . . . . .        | 23 |
| 4 | Tabella test di sistema . . . . .                                       | 26 |
| 5 | Risultati indice di $BV$ e $SV$ - Revisione dei Requisiti . . . . .     | 32 |
| 6 | Risultati indice di $Gulpease_G$ - Revisione dei Requisiti . . . . .    | 33 |
| 7 | Risultati indice di $BV$ e $SV$ - Revisione di Progettazione . . . . .  | 34 |
| 8 | Risultati indice di $Gulpease_G$ - Revisione di Progettazione . . . . . | 35 |



## Elenco delle figure

|   |   |    |
|---|---|----|
| 1 | Modello <i>ISO/IEC 15504<sub>G</sub></i> – <i>SPICE<sub>G</sub></i> . . . . . | 13 |
| 2 | <i>Ciclo di Deming<sub>G</sub></i> - <i>PDCA<sub>G</sub></i> . . . . .        | 15 |
| 3 | Modello <i>ISO/IEC 9126<sub>G</sub></i> . . . . .                             | 16 |
| 4 | <i>V-Model<sub>G</sub></i> per il test software . . . . .                     | 19 |

## 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<sub>G</sub>* per *Node.js<sub>G</sub>*, compatibile con l'utilizzo standard dei *middleware<sub>G</sub>* di *Express<sub>G</sub>* in versione 4.x. Il *framework<sub>G</sub>* prodotto dovrà permettere la realizzazione rapida di *client<sub>G</sub>* 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\_ver3.0.0* 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<sub>G</sub>*).

### 1.4 Riferimenti

#### 1.4.1 Normativi

- *PianoDiProgetto\_ver3.0.0*;
- *NormeDiProgetto\_ver2.0.0*;
- Capitolato C3  
<http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C3.pdf>

#### 1.4.2 Informativi

- Indice di *Gulpease<sub>G</sub>*:  
[http://www.http://it.wikipedia.org/wiki/Indice\\_Gulpease](http://www.http://it.wikipedia.org/wiki/Indice_Gulpease);
- Slide insegnamento *Ingegneria del Software<sub>G</sub>* Mod. A:  
<http://www.math.unipd.it/~tullio/IS-1/2014/>;
- Standard ISO<sub>G</sub>/IEC<sub>G</sub>9126:  
[http://it.wikipedia.org/wiki/ISO/IEC\\_9126](http://it.wikipedia.org/wiki/ISO/IEC_9126);
- Standard ISO<sub>G</sub>/IEC<sub>G</sub>15504:  
[http://en.wikipedia.org/wiki/ISO/IEC\\_15504](http://en.wikipedia.org/wiki/ISO/IEC_15504);
- *PDCA<sub>G</sub>* – *Ciclo di Deming<sub>G</sub>*:  
<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<sub>G</sub>/IEC<sub>G</sub>15504*, conosciuto anche con il nome di *SPICE<sub>G</sub>*. 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<sub>G</sub>*, conosciuto anche come *PDCA<sub>G</sub>*. Quest'ultimo mette a disposizione un iter per il controllo dei processi per garantire un miglioramento continuo durante tutto il loro *ciclo di vita<sub>G</sub>*.

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<sub>G</sub>/IEC<sub>G</sub>9126*, 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<sub>G</sub>* 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\_ver3.0.0*. 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 2.8.2.

### 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 2.8.1.

## 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\_ver2.0.0*. 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\_ver3.0.0*, 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\_ver2.0.0* sui documenti e sui processi attuati. Il resoconto di tali attività, precedenti la Revisione dei Requisiti, è riportato in appendice *D.1 - 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\_ver2.0.0*. Il resoconto di tali attività, precedenti la Revisione di Progettazione, è riportato in appendice *D.2 - 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\_ver2.0.0*. Viene verificato inoltre che i prodotti e i processi attuati nell'attività di progettazione dell'architettura seguano le procedure descritte nelle *NormeDiProgetto\_ver2.0.0*, 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.2 - 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\_ver2.0.0*. Viene verificato che i prodotti e i processi attuati nelle attività di progettazione di dettaglio e di codifica seguano le procedure descritte nelle *NormeDiProgetto\_ver2.0.0*. È inoltre garantito il risultato positivo dei test pianificati nella fase precedente, in conformità con le metriche definite nella sezione 2.8.1: *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\_ver2.0.0*. 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\_ver3.0.0*, 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\_ver2.0.0*.

## 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\_ver2.0.0*.

## 2.7 Tecniche di Analisi

### 2.7.1 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 due differenti metodologie che usualmente vengono applicate in maniera complementare.

#### 2.7.1.1 Inspection

L'attuazione di questa procedura implica un'analisi mirata solo di alcune sezioni del documento o del codice, cioè quelle che si ritiene probabili fonti del maggior numero di errori; risulta essere quindi molto più rapida del *Walkthrough*.

Essendo eseguita in modo mirato, come prima cosa deve avvenire una pianificazione

di quali parti andare a sottoporre alla verifica, poi deve essere definita una lista di controllo e solo successivamente si può passare all'attività di verifica vera e propria. 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 anomalie riscontrate o delle correzioni apportate, secondo quanto definito nel documento *NormeDiProgetto\_ver2.0.0*.

### 2.7.1.2 Walkthrough

Con questo metodo si va ad eseguire una lettura a largo spettro, 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 si rende necessario effettuare una verifica 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 di verifica per le fasi successive.

Essendo necessario un controllo completo dell'intero documento è necessario un impiego di risorse maggiore. Terminata la fase di ricerca dei possibili errori verrà effettuata una discussione per esporre i difetti trovati ed organizzare la correzione. Solo al termine di quest'ultima si procederà all'applicazione effettiva dei provvedimenti decisi. Anche in questo caso è necessario che venga presa nota delle anomalie riscontrate o delle correzioni apportate, secondo quanto definito nel documento *NormeDiProgetto\_ver2.0.0*.

### 2.7.2 Analisi dinamica

A differenza dell'analisi statica, l'analisi dinamica viene svolta solamente sul prodotto software e necessita della sua esecuzione. L'analisi verrà effettuata mediante lo svolgimento sul codice di test appositamente predisposti per individuare possibili malfunzionamenti o errori in fase di progettazione. Affinché l'analisi sia efficace, risulta necessario che i test siano ripetibili; questo è fondamentale, poiché 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 è quindi necessario definire a priori:

- **Ordine e procedure di esecuzione:** l'ordine e le modalità con cui i test devono essere eseguiti, e i metodi necessari all'analisi del risultato;
- **Relazione Input/Output:** gli input significativi per l'esecuzione e gli output attesi;
- **Ambiente di esecuzione:** il sistema hardware e software sul quale saranno effettuati questi test.

Sono stati individuati quindi cinque tipi di test: *test di unità*, *test di integrazione*, *test di regressione*, *test di sistema*, *test di accettazione*.

#### 2.7.2.1 Test di unità

Tramite i test di unità verrà controllata ogni singola componente creata per il prodotto. Questi test prevedono l'utilizzo di appositi *driver<sub>G</sub>*, *stub<sub>G</sub>* e *logger<sub>G</sub>*.

Lo scopo è quello di verificare il corretto funzionamento di ogni singola unità che compone il sistema e di eliminare i possibili errori commessi nel corso della programmazione. Durante questi test vengono rilevati circa i  $\frac{2}{3}$  dei difetti presenti. Con unità si intende 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.

#### **2.7.2.2 Test di integrazione**

Tramite i test di integrazione verrà verificata la combinazione di due o più unità al fine di permettere corrette aggiunte incrementali al prodotto software. Con questo approccio i possibili malfunzionamenti saranno facilmente tracciabili, in quanto rende 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/o realizzate da terzi.

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

#### **2.7.2.3 Test di regressione**

Tramite i test di regressione verrà verificato, se necessario, che eventuali modifiche apportate a una o più componenti non pregiudichino il funzionamento delle componenti già verificate e dipendenti da queste ultime, che non sono state modificate. Per capire quali test di unità e/o integrazione debbano essere ripetuti a causa della modifica apportata è necessario riferirsi al tracciamento.

#### **2.7.2.4 Test di sistema**

Tramite i test di sistema verrà verificato che il comportamento dinamico del sistema sia conforme ai requisiti software definiti nel documento *AnalisiRequisiti\_ver3.0.0*, e completo rispetto ad essi.

#### **2.7.2.5 Test di accettazione**

Il test di accettazione verrà svolto in presenza del Proponente, e sarà il collaudo ufficiale del prodotto software finito. L'esito positivo dello stesso comporterà il rilascio ufficiale del prodotto.

### **2.8 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<sub>G</sub>* 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.

### 2.8.1 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.

#### 2.8.1.1 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:**  $1 \rightarrow 3$ .

#### 2.8.1.2 Numero di parametri per metodo

Un elevato numero di parametri in input potrebbe portare ad un rapido riempimento dello *stack<sub>G</sub>*. 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:**  $0 \rightarrow 4$ .

#### 2.8.1.3 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:**  $1 \rightarrow 10$ .

#### 2.8.1.4 Linee per metodo

Rappresenta il numero di *statement<sub>G</sub>* 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:**  $0 \rightarrow 30$ .

### 2.8.1.5 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.

### 2.8.1.6 Chiamate innestate di metodi

In congiunzione all'elevato numero di parametri potrebbe portare ad un rapido riempimento dello *stack<sub>G</sub>*, 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:**  $0 \rightarrow 5$ .

### 2.8.1.7 Validazione *HTML<sub>G</sub>*

Il codice *HTML<sub>G</sub>* inserito per la creazione della pagina, su cui andare ad inserire i grafici, dovrà superare senza errori il test di validazione fornito dal *W3C<sub>G</sub>*. 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:**  $0 \rightarrow 0$ .

### 2.8.1.8 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%.

### 2.8.1.9 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:**  $0 \rightarrow 10$ .

## 2.8.2 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\_ver3.0.0*.

### 2.8.2.1 *Budget Variance<sub>G</sub>* (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\_ver3.0.0* di conseguenza.

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

### 2.8.2.2 *Schedule Variance<sub>G</sub>* (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:**  $EV \geq 0$ .

## 2.8.3 Metriche per i documenti

Come analisi sui documenti redatti, verrà verificato l'indice di *Gulpease<sub>G</sub>*, 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<sub>G</sub>*. Il gruppo **FlameTech Inc.**, dopo una valutazione di entrambe le metriche ha deciso di optare per l'utilizzo dell'indice di *Gulpease<sub>G</sub>*, 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.

### 2.8.3.1 Indice di *Gulpease<sub>G</sub>*

Rispetto ad altre metriche, l'indice di *Gulpease<sub>G</sub>* 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.

## 3 Gestione della revisione

### 3.1 Gestione anomalie riscontrate

Se durante la fase di verifica i *Verificatori* dovessero incontrare una o più anomalie, dovranno provvedere ad aprire un *ticket<sub>G</sub>* usando il sistema *Redmine<sub>G</sub>*. Per maggiori dettagli consultare la sezione dedicata del documento *NormeDiProgetto\_ver2.0.0*.

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.

## Appendici

### A Qualità di Processo

#### A.1 Standard *ISO/IEC 15504<sub>G</sub>*

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<sub>G</sub>*. Questo standard mette a disposizione degli strumenti per valutare il grado di maturità dei processi.

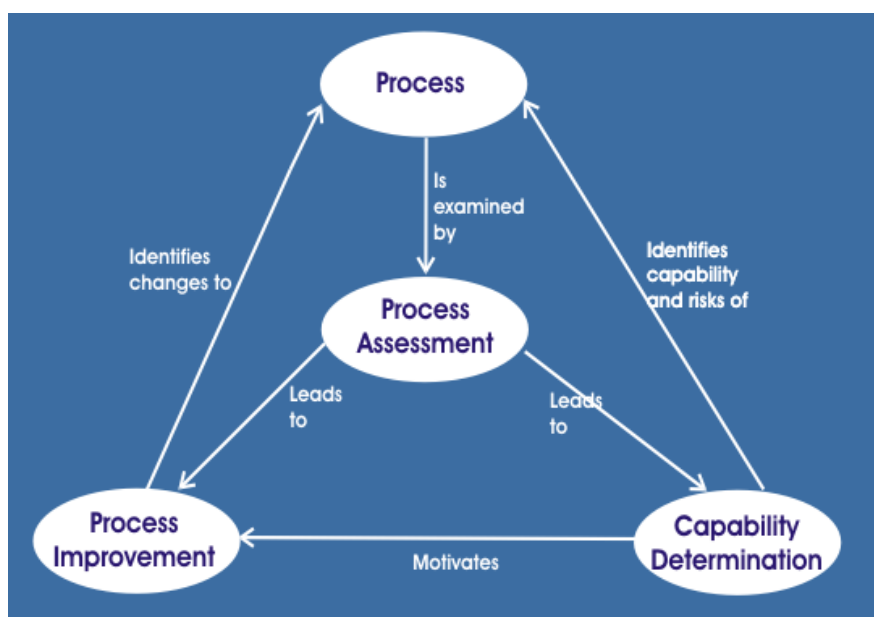


Figura 1: Modello *ISO/IEC 15504<sub>G</sub>* – *SPICE<sub>G</sub>*

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:

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%).

## A.2 *Ciclo di Deming*

Il *ciclo di Deming* o *PDCA<sub>G</sub>*, si suddivide nelle seguenti quattro fasi:



Figura 2: *Ciclo di Deming<sub>G</sub> - PDCA<sub>G</sub>*

- **Plan:** fase di pianificazione, in cui stabilire gli obiettivi, le scadenze, le responsabilità e i processi necessari per fornire risultati in accordo con i risultati attesi;
- **Do:** fase di attuazione del piano, in cui eseguire il processo e creare il prodotto;
- **Check:** fase di controllo in cui studiare i risultati effettivi (misurati e raccolti in Do) e confrontarli con i risultati attesi (obiettivi del Plan);
- **Act:** fase in cui applicare azioni correttive sulle differenze tra i risultati della verifica e quelli previsti, nell'ottica del miglioramento continuo.

## B Qualità di Prodotto

### B.1 Standard *ISO/IEC 9126<sub>G</sub>*

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

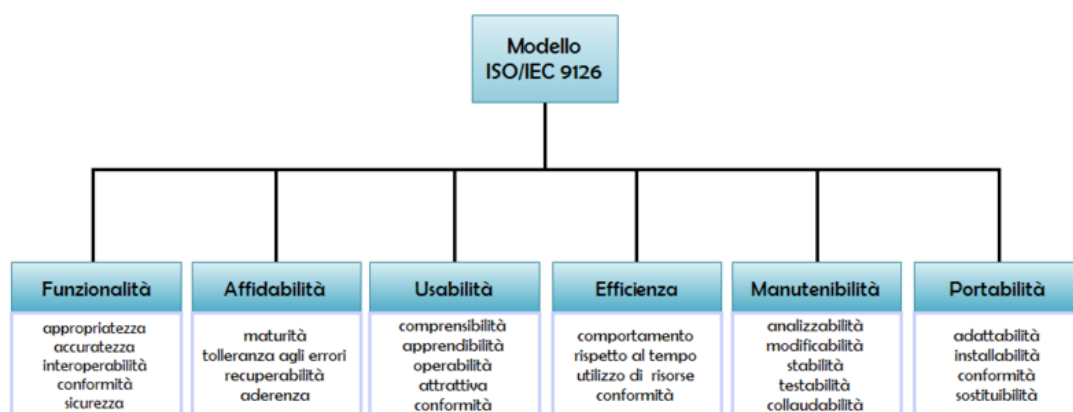


Figura 3: Modello *ISO/IEC 9126<sub>G</sub>*

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;
- **Replaceability:** capacità del prodotto software di essere utilizzato in sostituzione di un altro per svolgere gli stessi compiti nello stesso ambiente;
- **Portability Compliance:** capacità del prodotto software di aderire a standard, e regolamenti in materia di portabilità.



## C Pianificazione 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\_ver3.0.0*.

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.

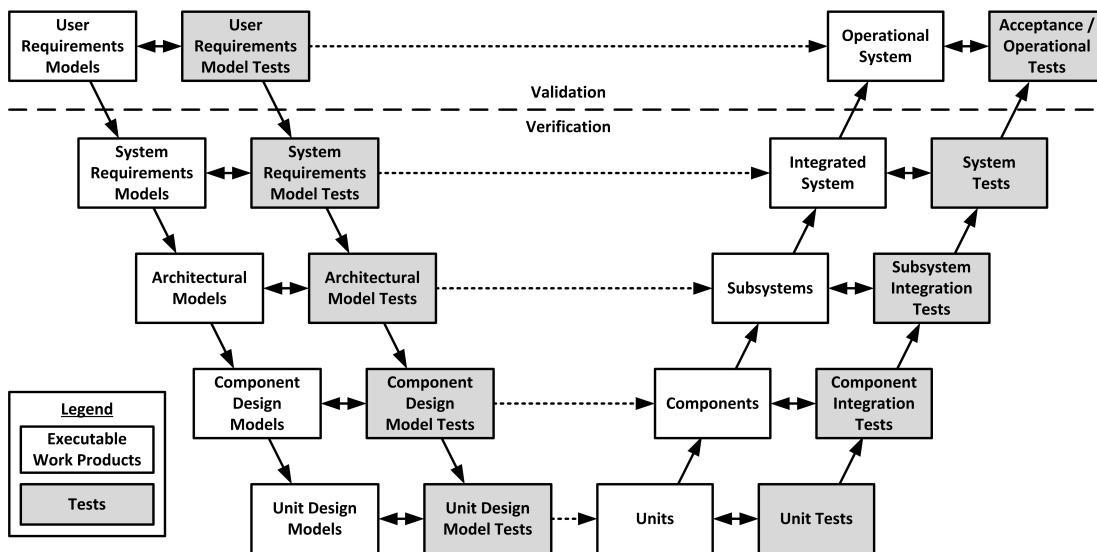


Figura 4: *V-Model<sub>G</sub>* per il test software

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<sub>G</sub>*, 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.Back-end         | Verificare che le componenti DeveloperProject e Lib si integrino correttamente tra loro permettendo la creazione di istanze di Norris correttamente funzionanti  | NT    | Norris::<br>Back-end                       |
| TI.DeveloperProject | Verificare che la componente DeveloperProject permetta di creare un nuovo progetto Norris e di configurarne le caratteristiche   | NT    | Norris::<br>Back-end::<br>DeveloperProject |
| TI.Lib              | Verificare che la libreria del <i>framework<sub>G</sub></i> 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::<br>Back-end::<br>Lib              |

| Test                 | Descrizione   | Stato | Componente   |
|----------------------|---|-------|--|
| TI.BusinessLayer     | Verificare che venga gestito correttamente il componente BusinessLayer relativo al <i>design pattern<sub>G</sub> Three Tier Architecture<sub>G</sub></i> , in particolare che venga gestita correttamente la comunicazione tra le componenti PresentationLayer e DataLayer                    | NT    | Norris::<br>Back-end::<br>Lib::<br>BusinessLayer     |
| TI.DataLayer         | Verificare che venga gestito correttamente il componente DataLayer relativo al <i>design pattern<sub>G</sub> Three Tier Architecture<sub>G</sub></i> , in particolare che vengano gestite correttamente le informazioni relative alle pagine ed ai grafici                                    | NT    | Norris::<br>Back-end::<br>Lib::<br>DataLayer         |
| TI.PresentationLayer | Verificare che venga gestito correttamente il componente PresentationLayer relativo al <i>design pattern<sub>G</sub> Three Tier Architecture<sub>G</sub></i> , in particolare che vengano gestite correttamente le richieste <i>HTTP<sub>G</sub></i> effettuate dai <i>client<sub>G</sub></i> | NT    | Norris::<br>Back-end::<br>Lib::<br>PresentationLayer |
| TI.Front-end         | 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::<br>Front-end                                |
| TI.Controller        | Verificare che venga gestito correttamente il componente Controller relativo al <i>design pattern<sub>G</sub> MVW<sub>G</sub></i> , in particolare che vengano creati correttamente i grafici e che la pagina visualizzata dal <i>client<sub>G</sub></i> sia gestita correttamente            | NT    | Norris::<br>Front-end::<br>Controller                |
| TI.Model             | Verificare che venga gestito correttamente il componente Model relativo al <i>design pattern<sub>G</sub> MVW<sub>G</sub></i> , in particolare che vengano mantenuti e gestiti correttamente i dati dei grafici presenti sulla pagina visualizzata dal <i>client<sub>G</sub></i>               | NT    | Norris::<br>Front-end::<br>Model                     |

| Test          | Descrizione   | Stato | Componente                         |
|---------------|---|-------|------------------------------------|
| TI.Service    | Verificare che venga gestito correttamente il componente Service relativo al <i>design pattern<sub>G</sub> MVW<sub>G</sub></i> , in particolare che vengano gestite correttamente le notifiche di aggiornamento da parte del <i>server<sub>G</sub></i>  | NT    | Norris::<br>Front-end::<br>Service |
| TI.View       | Verificare che venga gestito correttamente il componente View relativo al <i>design pattern<sub>G</sub> MVW<sub>G</sub></i> , in particolare che il sistema gestisca correttamente i <i>template<sub>G</sub></i> delle pagine <i>HTML<sub>G</sub></i> e il codice <i>JavaScript<sub>G</sub></i> | NT    | Norris::<br>Front-end::<br>View    |
| TI.AndroidApp | Verificare che le componenti Layouts, Activities e AppModel si integrino correttamente tra loro garantendo un corretto funzionamento complessivo dell'applicativo <i>Android<sub>G</sub></i>  | NT    | AndroidApp                         |
| TI.Activities | Verificare che venga gestito correttamente il componente Controller relativo al <i>design pattern<sub>G</sub> MVC<sub>G</sub></i> , in particolare che venga gestita correttamente la logica delle varie attività dell'applicativo  | NT    | AndroidApp::<br>Activities         |
| TI.AppModel   | Verificare che venga gestito correttamente il componente Model relativo al <i>design pattern<sub>G</sub> MVC<sub>G</sub></i> , in particolare che vengano mantenuti e gestiti correttamente i dati necessari al funzionamento dell'applicativo  | NT    | AndroidApp::<br>AppModel           |
| TI.Layouts    | Verificare che venga gestito correttamente il componente View relativo al <i>design pattern<sub>G</sub> MVC<sub>G</sub></i> , in particolare che vengano visualizzati i layout corretti per le varie funzionalità dell'applicativo  | NT    | AndroidApp::<br>Layouts            |

Tabella 2: Tabella test di integrazione

### C.1.2 Tracciamento componenti - test di integrazione

| Componente                               | Test integrazione    |
|--|----------------------|
| Norris                                   | TI.Norris            |
| Norris::Back-end                         | TI.Back-end          |
| Norris::Back-end::DeveloperProject       | TI.DeveloperProject  |
| Norris::Back-end::Lib                    | TI.Lib               |
| Norris::Back-end::Lib::BusinessLayer     | TI.BusinessLayer     |
| Norris::Back-end::Lib::DataLayer         | TI.DataLayer         |
| Norris::Back-end::Lib::PresentationLayer | TI.PresentationLayer |
| Norris::Front-end                        | TI.Front-end         |
| Norris::Front-end::Controller            | TI.Controller        |
| Norris::Front-end::Model                 | TI.Model             |
| Norris::Front-end::Service               | TI.Service           |
| Norris::Front-end::View                  | TI.View              |
| 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\_ver3.0.0*.

### C.2.1 Descrizione dei test di sistema

| Test   | Descrizione  | Stato | Requisito |
|--------|--|-------|-----------|
| TSF1   | Verificare che il <i>framework<sub>G</sub></i> renda disponibili delle <i>API<sub>G</sub></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<sub>G</sub></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 e di assegnargli le dimensioni di visualizzazione  | NT    | RAF1.3    |
| TSF1.4 | Verificare che il <i>framework<sub>G</sub></i> fornisca un sistema di log per il tracciamento degli errori   | NT    | RBF1.4    |
| 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<sub>G</sub></i> permetta di configurare vari parametri sui grafici  | NT    | RAF4      |
| TSF5   | Verificare che sia disponibile una <i>dashboard<sub>G</sub></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_ver2.0.0</i>  | NT    | RAQ1      |

| Test   | Descrizione  | Stato | Requisito |
|--------|--|-------|-----------|
| TSQ2   | Verificare che Norris <i>framework<sub>G</sub></i> sia distribuito con <i>Licenza MIT<sub>G</sub></i> e che menzioni il contributo di CoffeeStrap  | NT    | RAQ2      |
| 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<sub>G</sub></i>   | NT    | RAQ4      |
| TSV1   | Verificare che la componente <i>server<sub>G</sub></i> sia stata realizzata utilizzando <i>Node.js<sub>G</sub></i>   | NT    | RAV1      |
| TSV2   | Verificare che il sistema utilizzi la versione 4.x di <i>Express<sub>G</sub></i> per la realizzazione dell'infrastruttura per la componente <i>server<sub>G</sub></i>                                      | NT    | RAV2      |
| TSV3   | Verificare che il sistema utilizzi <i>Socket.io<sub>G</sub></i> per la componente <i>WebSocket<sub>G</sub></i> che realizza le notifiche <i>push<sub>G</sub></i>   | NT    | RAV3      |
| TSV4   | Verificare che il sistema lato <i>client<sub>G</sub></i> sia completamente compatibile con il browser <i>Chrome<sub>G</sub></i> versione 38.0.X o superiori  | NT    | RAV4      |
| TSV5   | Verificare che il sistema lato <i>client<sub>G</sub></i> sia completamente compatibile con il browser <i>Firefox<sub>G</sub></i> versione 32.X o superiori   | NT    | RAV5      |
| TSV6   | Verificare che il progetto sia caricato su <i>GitHub<sub>G</sub></i> e permetta l'utilizzo delle <i>issue<sub>G</sub></i> per la segnalazione di <i>bug<sub>G</sub></i>                                    | NT    | RAV6      |
| TSV7   | Verificare che sia stato effettuato il <i>deployment<sub>G</sub></i> su <i>Heroku<sub>G</sub></i>  | NT    | RCV7      |
| TSV8   | Verificare che sia possibile installare il <i>framework<sub>G</sub></i> tramite l'utilizzo del <i>repository<sub>G</sub></i> <i>npm<sub>G</sub></i> di <i>Node.js<sub>G</sub></i> , <i>npm<sub>G</sub></i> | NT    | RAV8      |
| TSV8.1 | Verificare che il <i>framework<sub>G</sub></i> importi automaticamente le librerie esterne necessarie al funzionamento quali <i>Express<sub>G</sub></i> e <i>Socket.io<sub>G</sub></i>                     | NT    | RBV8.1    |

| Test   | Descrizione   | Stato | Requisito |
|--------|---|-------|-----------|
| TSV8.2 | Verificare che il processo di installazione crei correttamente il <i>template<sub>G</sub></i> e le sottocartelle necessarie per il funzionamento del <i>framework<sub>G</sub></i> | NT    | RAV8.2    |
| TSV9   | Verificare che l'applicazione <i>Android<sub>G</sub></i> sia compatibile con la versione 4.1 o superiori di <i>Android<sub>G</sub></i>  | NT    | RAV9      |

Tabella 4: Tabella 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\_ver3.0.0*.

#### 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  $framework$  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<sub>G</sub>* (TV5.1.1);
  - Verificare che sia possibile creare un grafico di tipo *Line Chart<sub>G</sub>* (TV5.1.2);
  - Verificare che sia possibile creare un grafico di tipo *Map Chart<sub>G</sub>* (TV5.1.3);
  - Verificare che sia possibile creare un grafico di tipo *Table<sub>G</sub>* (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<sub>G</sub>* per la visualizzazione dei grafici attivi.

Allo sviluppatore è richiesto di:

- Accedere all'indirizzo della *dashboard<sub>G</sub>* (TV6.1);
- Verificare che la *dashboard<sub>G</sub>* visualizzi i grafici attivi sul *server<sub>G</sub>* (TV6.2).

### C.3.7 Test TV7

L'utente vuole verificare che sia possibile visualizzare grafici tramite l'applicazione *Android<sub>G</sub>*.

All'utente è richiesto di:

- Verificare che sia possibile inserire un indirizzo *URL<sub>G</sub>* 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<sub>G</sub>* (TV8.14);
- Verificare che sia possibile ordinare con criterio alfanumerico, crescente o decrescente, gli elementi di un grafico di tipo *Table<sub>G</sub>* 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<sub>G</sub>* 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<sub>G</sub>* 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 formato di visualizzazione dei dati (TV9.2);
- Verificare che sia possibile modificare i colori dei dati (TV9.3);
- Verificare che sia possibile modificare il colore per ogni cella di un grafico di tipo *Table<sub>G</sub>* (TV9.4);
- Verificare che sia possibile modificare l'opzione di visualizzazione della legenda di un grafico (TV9.5);
- Verificare che sia possibile modificare la posizione della legenda di un grafico (TV9.6);
- Verificare che sia possibile modificare l'opzione di visualizzazione della griglia degli assi (TV9.7);
- Verificare che sia possibile modificare l'orientamento delle barre di un grafico di tipo *Bar Chart<sub>G</sub>* tra verticale e orizzontale (TV9.8);
- Verificare che sia possibile modificare il punto centrale della mappa di un grafico di tipo *Map Chart<sub>G</sub>* (TV9.9);
- Verificare che sia possibile modificare le dimensioni dell'area di mappa visualizzata di un grafico di tipo *Map Chart<sub>G</sub>* (TV9.10);
- Verificare che sia possibile modificare l'opzione di visualizzazione dei bordi di un grafico di tipo *Table<sub>G</sub>* (TV9.11);
- Verificare che sia possibile modificare l'opzione di ordinamento con criterio alfanumerico, crescente o decrescente, degli elementi di un grafico di tipo *Table<sub>G</sub>* rispetto ad una colonna (solo con aggiornamento in place) (TV9.12);
- Verificare che sia possibile scegliere se l'aggiunta di nuovi elementi ad un grafico di tipo *Table<sub>G</sub>* avviene in testa o in coda (solo con aggiornamento stream) (TV9.13);
- Verificare che sia possibile modificare il massimo numero di elementi di un grafico di tipo *Table<sub>G</sub>* visualizzati contemporaneamente (TV9.14).

### C.3.10 Test TV10

L'utente vuole verificare che sia possibile visualizzare una *dashboard<sub>G</sub>* 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<sub>G</sub>* (TV10.1);
- Selezionare una linea degli autobus (TV10.2);
- Verificare che nel grafico appaiano i dati degli autobus appartenenti alla linea selezionata (TV10.3).

## D Risultati della verifica

### D.1 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\_ver2.0.0*. 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\_ver2.0.0*. 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<sub>G</sub>*.

Grazie al continuo controllo sui processi attuati, sulla base di quanto descritto nel documento *NormeDiProgetto\_ver2.0.0*, 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\_ver3.0.0*.

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 5: 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<sub>G</sub>* che sono stati inseriti durante il periodo di pianificazione tutte le attività sono state svolte rispettando i tempi pianificati come indicato nel documento *PianoDiProgetto\_ver3.0.0*;
- Il valore del *BV<sub>G</sub>* 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<sub>G</sub>*. 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 6: Risultati indice di *Gulpease<sub>G</sub>* - 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.1.1 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 apportato 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<sub>G</sub>* 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<sub>G</sub>* e i requisiti riguardanti l'applicazione *Android<sub>G</sub>* 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.2 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\_ver2.0.0*. In particolare si è ricorsi all'utilizzo della tecnica walkthrough per il documento *SpecificaTecnica\_ver1.0.0* 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\_ver2.0.0*. 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<sub>G</sub>*.

Grazie al continuo controllo sui processi attuati, sulla base di quanto descritto nel documento *NormeDiProgetto\_ver2.0.0*, 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\_ver3.0.0*.

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 7: 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<sub>G</sub>* che sono stati inseriti durante il periodo di pianificazione tutte le attività sono state svolte rispettando i tempi pianificati come indicato nel documento *PianoDiProgetto\_ver3.0.0*;
- Il valore del *BV<sub>G</sub>* 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<sub>G</sub>*. 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 8: Risultati indice di *Gulpease<sub>G</sub>* - 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.