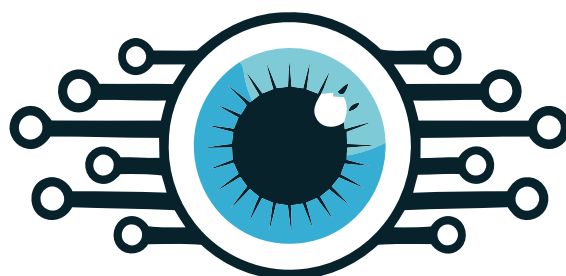


PREMI



Pragma

Piano di Qualifica

Informazioni sul documento

Versione	3.0.0
Redazione	Massimiliano Baruffato Giacomo Manzoli Gianmarco Midena Stefano Munari
Verifica	Daniele Marin Fabio Vedovato
Approvazione	Gianmarco Midena
Uso	Esterno
Distribuzione	Pragma
Destinato a	Prof. Vardanega Tullio Prof. Cardin Riccardo Zucchetti S.p.A.

Sommario

Strategie di verifica adottate dal gruppo Pragma volte alla perseguimento costante di requisiti qualitativi per il progetto Premi

A.A. 2014-15

pragma.swe@gmail.com

Diario delle modifiche

Versione	Descrizione modifica	Autore	Ruolo	Data
3.0.0	<i>Approvazione documento</i>	Gianmarco Midena	Responsabile	2015-05-22
2.15.1	<i>Verifica del documento</i>	Daniele Marin	Verificatore	2015-05-22
2.15.0	<i>Correzione imprecisioni sui Test di Unità per la parte Front-End sezioni “Specificazione dei test” (sezione 4) e “Tracciamenti dei test” (sezione 5)</i>	Giacomo Manzoli	Progettista	2015-05-22
2.14.1	<i>Verifica del documento</i>	Fabio Vedovato	Verificatore	2015-05-21
2.14.0	<i>Aggiornamento sezione “Resoconto attività di verifica” (appendice A)</i>	Gianmarco Midena	Analista	2015-05-21
2.13.0	<i>Aggiornamento sezioni “Specificazione dei test” (sezione 4) e “Tracciamenti dei test” (sezione 5) aggiungendo i Test di Unità per la parte Front-End</i>	Giacomo Manzoli	Progettista	2015-05-20
2.12.0	<i>Aggiornamento sezioni “Specificazione dei test” (sezione 4) e “Tracciamenti dei test” (sezione 5) aggiungendo i Test di Unità per la parte Back-End</i>	Stefano Munari	Progettista	2015-05-19
2.11.0	<i>Aggiunta metrica “Indice di manutenibilità - Non Accettabilità” (sezione 2.7.2.13)</i>	Gianmarco Midena	Analista	2015-05-18
2.10.0	<i>Aggiunta metrica “Halstead Difficulty per-function - Non Accettabilità” (sezione 2.7.2.9)</i>	Gianmarco Midena	Analista	2015-05-18
2.9.0	<i>Aggiunta metrica “Complessità Ciclomantica - Non Accettabilità” (sezione 2.7.2.3)</i>	Gianmarco Midena	Analista	2015-05-18
2.8.0	<i>Aggiornata metrica “Copertura dei test” (sezione 2.11.2.1)</i>	Gianmarco Midena	Analista	2015-05-18
2.7.1	<i>Verifica del documento</i>	Fabio Vedovato	Verificatore	2015-05-15
2.7.0	<i>Aggiunta metrica “Parametri per metodo - Non Accettabilità” (sezione 2.6.2.4)</i>	Massimiliano Baruffato	Analista	2015-05-14
2.6.0	<i>Aggiunta metrica “Metodi per classe - Non Accettabilità” (sezione 2.6.2.2)</i>	Massimiliano Baruffato	Analista	2015-05-14
2.5.0	<i>Aggiunta metrica “SFOUT - Non Accettabilità” (sezione 2.5.2.4)</i>	Massimiliano Baruffato	Analista	2015-05-14
2.4.0	<i>Aggiunta metrica “SFIN - Ottimalità” (sezione 2.5.2.2)</i>	Massimiliano Baruffato	Analista	2015-05-14

Versione	Descrizione modifica	Autore	Ruolo	Data
2.3.0	Aggiornata metrica “Numero di metodi” in “Numero di metodi per classe”, corrette imprecisioni e modificati i suoi valori di riferimento (sezione 2.6.2.1)	Massimiliano Baruffato	Analista	2015-05-11
2.2.0	Corretta imprecisione metrica “Structural Fan-Out” (sezione 2.5.2.3)	Massimiliano Baruffato	Analista	2015-05-11
2.1.0	Corretta sezione “Pianificazione dei test” in “Specifica dei test” (sezione 4)	Massimiliano Baruffato	Analista	2015-05-11
2.0.0	Approvazione documento	Massimiliano Baruffato	Responsabile	2015-04-19
1.38.1	Verifica del documento	Giacomo Manzoli	Verificatore	2015-04-16
1.38.0	Stesura sezioni “Pianificazione dei test” (sezione 4) e “Tracciamenti dei test” (sezione 5)	Daniele Marin	Progettista	2015-04-02
1.37.0	Aggiornamento sezione “Resoconto attività di verifica” (appendice A)	Massimiliano Baruffato	Analista	2015-03-25
1.36.2	Approvazione del documento	Andrea Ongaro	Responsabile	2015-03-23
1.36.1	Verifica del documento	Gianmarco Midena	Verificatore	2015-03-23
1.36.0	Corrette ambiguità nella spiegazione del termine “Efficienza” e negli obiettivi di qualità (sezione 3.4)	Fabio Vedovato	Analista	2015-03-19
1.35.0	Corrette ambiguità negli obiettivi di qualità di usabilità (sezione 3.3)	Fabio Vedovato	Analista	2015-03-19
1.34.0	Corrette ambiguità nella spiegazione del termine “Affidabilità” e negli obiettivi di qualità (sezione 3.2)	Fabio Vedovato	Analista	2015-03-19
1.33.0	Corrette ambiguità nella spiegazione del termine “Funzionalità” e negli obiettivi di qualità (sezione 3.1)	Fabio Vedovato	Analista	2015-03-19
1.32.0	Aggiunta metrica “Indice di manutenibilità” (sezione 2.7.2.12)	Fabio Vedovato	Analista	2015-03-18
1.31.0	Aggiunta metrica “Halstead Effort per-function” (sezione 2.7.2.11)	Fabio Vedovato	Analista	2015-03-18
1.30.0	Aggiunta metrica “Halstead Volume per-function” (sezione 2.7.2.10)	Fabio Vedovato	Analista	2015-03-18

Versione	Descrizione modifica	Autore	Ruolo	Data
1.29.0	Aggiunta metrica “Halstead Difficulty per-function” (sezione 2.7.2.8)	Fabio Vedovato	Analista	2015-03-18
1.28.0	Aggiunta metrica “Dipendenze” (sezione 2.7.2.7)	Fabio Vedovato	Analista	2015-03-18
1.27.0	Aggiornata metrica “Efficienza di gestione dei rischi” (sezione 2.3.2.2)	Fabio Vedovato	Analista	2015-03-17
1.26.0	Aggiornata metrica “Errori di individuazione termini di glossario” (sezione 2.1.2.3)	Fabio Vedovato	Analista	2015-03-17
1.25.0	Aggiornata metrica “Tempo di correzione incoerenze in PragmaDB” (sezione 2.1.2.2)	Fabio Vedovato	Analista	2015-03-17
1.24.0	Corretta ambiguità nella metrica “Disponibilità PragmaDB” (sezione 2.1.2.1)	Fabio Vedovato	Analista	2015-03-17
1.23.0	Corretta imprecisione obiettivi di qualità per il processo “Infrastructure Management Process (6.2.2)” (sezione 2.1)	Fabio Vedovato	Analista	2015-03-17
1.22.1	Verifica del documento	Gianmarco Midena	Verificatore	2015-03-16
1.22.0	Aggiunta sezione “Portabilità (6.6)” (sezione 3.6)	Fabio Vedovato	Analista	2015-03-11
1.21.0	Aggiunta sezione “Manutenibilità (6.5)” (sezione 3.5)	Fabio Vedovato	Analista	2015-03-11
1.20.0	Aggiunta sezione “Efficienza (6.4)” (sezione 3.4)	Fabio Vedovato	Analista	2015-03-10
1.19.0	Aggiunta sezione “Usabilità (6.3)” (sezione 3.3)	Fabio Vedovato	Analista	2015-03-09
1.18.0	Aggiunta sezione “Affidabilità (6.2)” (sezione 3.2)	Fabio Vedovato	Analista	2015-03-09
1.17.0	Aggiunta sezione “Funzionalità (6.1)” (sezione 3.1)	Fabio Vedovato	Analista	2015-03-09
1.16.0	Inizio stesura “Qualità di prodotto” (sezione 3)	Fabio Vedovato	Analista	2015-03-09
1.15.2	Approvazione del documento	Andrea Ongaro	Responsabile	2015-03-08
1.15.1	Verifica del documento	Gianmarco Midena	Verificatore	2015-03-07
1.15.0	Aggiunta sezione “Software Verification Process (7.2.4)” (sezione 2.11)	Fabio Vedovato	Analista	2015-03-06
1.14.0	Aggiunta sezione “Software Documentation Management Process (7.2.1)” (sezione 2.10)	Fabio Vedovato	Analista	2015-03-05

Versione	Descrizione modifica	Autore	Ruolo	Data
1.13.0	Aggiunta sezione "System/Software Qualification Testing Process (6.4.6 - 7.1.7)" (sezione 2.9)	Fabio Vedovato	Analista	2015-03-05
1.12.0	Aggiunta sezione "System/Software Integration Process (6.4.5 - 7.1.6)" (sezione 2.8)	Fabio Vedovato	Analista	2015-03-05
1.11.0	Aggiunta sezione "Software Construction Process (7.1.5)" (sezione 2.7)	Fabio Vedovato	Analista	2015-03-03
1.10.0	Aggiunta sezione "Software Detailed Design Process (7.1.4)" (sezione 2.6)	Fabio Vedovato	Analista	2015-03-03
1.9.0	Aggiunta sezione "System/Software Architectural Design Process (6.4.3 - 7.1.3)" (sezione 2.5)	Fabio Vedovato	Analista	2015-03-03
1.8.0	Aggiunta sezione "System/Software Requirements Analysis Process (6.4.2 - 7.1.2)" (sezione 2.4)	Fabio Vedovato	Analista	2015-03-02
1.7.0	Aggiunta sezione "Risk Management Process (6.3.4)" (sezione 2.3)	Fabio Vedovato	Analista	2015-03-02
1.6.0	Aggiunta sezione "Project Planning, Assessment & Control Process (6.3.1 - 6.3.2)" (sezione 2.2)	Fabio Vedovato	Analista	2015-02-28
1.5.0	Aggiunta sezione "Infrastructure Management Process (6.2.2)" (sezione 2.1)	Fabio Vedovato	Analista	2015-02-28
1.4.0	Inizio stesura "Qualità di processo" (sezione 2)	Fabio Vedovato	Analista	2015-02-27
1.3.0	Aggiornata sezione "Informativi" (sezione 1.4.2) inserendo nuovi riferimenti	Fabio Vedovato	Analista	2015-02-27
1.2.0	Stabilita nuova struttura del documento ed individuati contenuti chiave per decisioni E1-6 ed E1-7	Fabio Vedovato	Analista	2015-02-27
1.1.0	Spostate le sezioni "Risorse", "Strumenti", "Tecniche di analisi" e "Gestione amministrativa della revisione" nel documento Norme di Progetto v1.1.0	Fabio Vedovato	Analista	2015-02-27
1.0.0	Approvazione documento	Giacomo Manzoli	Responsabile	2015-01-18
0.4.1	Verifica documento	Fabio Vedovato	Verificatore	2015-01-18
0.4.0	Resoconto attività di verifica	Daniele Marin	Verificatore	2015-01-14

Versione	Descrizione modifica	Autore	Ruolo	Data
0.3.0	<i>Completata stesura documento</i>	Massimiliano Baruffato	Amministratore	2014-12-17
0.2.0	<i>Stesura strategie di verifica</i>	Andrea Ongaro	Analista	2014-12-16
0.1.0	<i>Stesura obiettivi di qualità</i>	Daniele Marin	Responsabile	2014-12-13
0.0.0	<i>Impostazione scheletro documento</i>	Andrea Ongaro	Analista	2014-12-10

Tabella 1: Diario delle modifiche.

Indice

1	Introduzione	12
1.1	Scopo del documento	12
1.2	Scopo del prodotto	12
1.3	Glossario	12
1.4	Riferimenti	12
1.4.1	Normativi	12
1.4.2	Informativi	12
2	Qualità di processo	13
2.1	Infrastructure Management Process (6.2.2)	13
2.1.1	Obiettivi di qualità	13
2.1.2	Metriche	14
2.1.2.1	Disponibilità <i>PragmaDB</i>	14
2.1.2.2	Tempo di correzione incoerenze in <i>PragmaDB</i>	14
2.1.2.3	Errori di individuazione termini di glossario	14
2.1.3	Strategie	14
2.2	Project Planning, Assessment & Control Process (6.3.1 - 6.3.2)	15
2.2.1	Obiettivi di qualità	15
2.2.2	Metriche	15
2.2.2.1	Schedule Variance	15
2.2.2.2	Budget Variance	15
2.2.3	Strategie	15
2.3	Risk Management Process (6.3.4)	16
2.3.1	Obiettivi di qualità	16
2.3.2	Metriche	16
2.3.2.1	Rischi non preventivati	16
2.3.2.2	Efficienza di gestione dei rischi	16
2.3.3	Strategie	16
2.4	System/Software Requirements Analysis Process (6.4.2 - 7.1.2)	16
2.4.1	Obiettivi di qualità	17
2.4.2	Metriche	17
2.4.2.1	Requisiti obbligatori soddisfatti	17
2.4.2.2	Requisiti desiderabili/facoltativi accettati soddisfatti	17
2.4.2.3	Requisiti desiderabili/facoltativi non accettati soddisfatti	17
2.4.3	Strategie	17
2.5	System/Software Architectural Design Process (6.4.3 - 7.1.3)	18
2.5.1	Obiettivi di qualità	18
2.5.2	Metriche	18
2.5.2.1	Structural Fan-In	18
2.5.2.2	SFIN - Ottimalità	18
2.5.2.3	Structural Fan-Out	18
2.5.2.4	SFOUT - Non Accettabilità	19
2.5.3	Strategie	19
2.6	Software Detailed Design Process (7.1.4)	19
2.6.1	Obiettivi di qualità	19
2.6.2	Metriche	19
2.6.2.1	Numero di metodi per classe	19
2.6.2.2	Metodi per classe - Non Accettabilità	20
2.6.2.3	Numero di parametri per metodo	20
2.6.2.4	Parametri per metodo - Non Accettabilità	20
2.6.3	Strategie	20

2.7	Software Construction Process (7.1.5)	20
2.7.1	Obiettivi di qualità	20
2.7.2	Metriche	21
2.7.2.1	Produttività di codifica	21
2.7.2.2	Complessità Ciclomatica	21
2.7.2.3	Complessità Ciclomatica - Non Accettabilità	21
2.7.2.4	Linee di commento su linee di codice	21
2.7.2.5	Chiamate annidate	22
2.7.2.6	Variabili inutilizzate	22
2.7.2.7	Dipendenze	22
2.7.2.8	Halstead Difficulty per-function	22
2.7.2.9	Halstead Difficulty per-function - Non Accettabilità	22
2.7.2.10	Halstead Volume per-function	23
2.7.2.11	Halstead Effort per-function	23
2.7.2.12	Indice di manutenibilità	23
2.7.2.13	Indice di manutenibilità - Non Accettabilità	23
2.7.3	Strategie	23
2.8	System/Software Integration Process (6.4.5 - 7.1.6)	23
2.8.1	Obiettivi di qualità	24
2.8.2	Metriche	24
2.8.2.1	Componenti integrate	24
2.8.3	Strategie	24
2.9	System/Software Qualification Testing Process (6.4.6 - 7.1.7)	24
2.9.1	Obiettivi di qualità	24
2.9.2	Metriche	24
2.9.2.1	Test di sistema superati	24
2.9.3	Strategie	25
2.10	Software Documentation Management Process (7.2.1)	25
2.10.1	Obiettivi di qualità	25
2.10.2	Metriche	25
2.10.2.1	Indice Gulpease	25
2.10.3	Strategie	25
2.11	Software Verification Process (7.2.4)	25
2.11.1	Obiettivi di qualità	25
2.11.2	Metriche	26
2.11.2.1	Copertura dei test	26
2.11.3	Strategie	26
3	Qualità di prodotto	27
3.1	Funzionalità (6.1)	27
3.1.1	Obiettivi di qualità	27
3.1.2	Metriche	27
3.1.2.1	Completezza dell'implementazione funzionale	27
3.1.2.2	Accuratezza rispetto alle attese	27
3.1.2.3	Controllo degli accessi	27
3.2	Affidabilità (6.2)	28
3.2.1	Obiettivi di qualità	28
3.2.2	Metriche	28
3.2.2.1	Densità di failure	28
3.2.2.2	Blocco di operazioni non corrette	28
3.3	Usabilità (6.3)	28
3.3.1	Obiettivi di qualità	28
3.3.2	Metriche	29
3.3.2.1	Comprensibilità delle funzioni offerte	29

3.3.2.2	Facilità di apprendimento delle funzionalità	29
3.3.2.3	Consistenza operativa in uso	29
3.3.2.4	Elementi personalizzabili	29
3.4	Efficienza (6.4)	29
3.4.1	Obiettivi di qualità	30
3.4.2	Metriche	30
3.4.2.1	Tempo di risposta	30
3.5	Manutenibilità (6.5)	30
3.5.1	Obiettivi di qualità	30
3.5.2	Metriche	30
3.5.2.1	Capacità di analisi di failure	30
3.5.2.2	Impatto delle modifiche	31
3.6	Portabilità (6.6)	31
3.6.1	Obiettivi di qualità	31
3.6.2	Metriche	31
3.6.2.1	Versioni dei browser supportate	31
3.6.2.2	Inclusione di funzionalità da altri prodotti	31
4	Specifica dei test	32
4.1	Tipi di test	32
4.2	Test di Validazione	32
4.3	Test di Sistema	48
4.4	Test di Integrazione	50
4.5	Test di Unità	53
5	Tracciamenti dei test	66
5.1	Tracciamento Test di Validazione-Requisiti	66
5.2	Tracciamento Requisiti-Test di Validazione	67
5.3	Tracciamento Test di Sistema-Requisiti	68
5.4	Tracciamento Requisiti-Test di Sistema	69
5.5	Tracciamento Test di Integrazione-Componenti	70
5.6	Tracciamento Componenti-Test di Integrazione	71
5.7	Tracciamento Test di Unità-Metodi	72
5.8	Tracciamento Metodi-Test di Unità	84
Appendice A	Resoconto attività di verifica	96
A.1	Fase di Ammissione al progetto	96
A.1.1	Analisi statica dei documenti	96
A.1.1.1	Esiti verifiche automatizzate	96
A.1.2	Considerazioni finali	97
A.2	Fase di Consolidamento dei requisiti	97
A.2.1	Analisi statica dei documenti	97
A.2.1.1	Esiti verifiche automatizzate	97
A.2.2	Considerazioni finali	97
A.3	Fase di Progettazione dell'architettura	98
A.3.1	Analisi statica dei documenti	98
A.3.1.1	Esiti verifiche automatizzate	98
A.3.2	Considerazioni finali	99
A.4	Fase di Consolidamento dell'architettura	99
A.4.1	Analisi statica dei documenti	99
A.4.1.1	Esiti verifiche automatizzate	100
A.4.2	Considerazioni finali	100
A.5	Fase di Realizzazione del prodotto	100
A.5.1	Analisi statica dei documenti	101

A.5.1.1	Esiti verifiche automatizzate	101
A.5.2	Soddisfacimento metriche	102
A.5.3	Considerazioni finali	103

Elenco delle figure

1	Albero di integrazione delle componenti	50
2	Diagramma di attività dei test di integrazione	51

Elenco delle tabelle

1	Diario delle modifiche.	5
2	Test di Validazione	47
3	Test di Sistema	49
4	Test di Integrazione	52
5	Test di Unità	65
6	Tracciamento Test di Validazione-Requisiti	66
7	Tracciamento Requisiti-Test di Validazione	67
8	Tracciamento Test di Sistema-Requisiti	68
9	Tracciamento Requisiti-Test di Sistema	69
10	Tracciamento Test di Integrazione-Componenti	70
11	Tracciamento Componenti-Test di Integrazione	71
12	Tracciamento Test di Unità-Metodi	83
13	Tracciamento Metodi-Test di Unità	95
14	Resoconto verifiche automatizzate - Fase di Ammissione al progetto	96
15	Resoconto verifiche automatizzate - Fase di Consolidamento dei requisiti	97
16	Resoconto verifiche automatizzate - Fase di Ammissione al progetto	98
17	Resoconto verifiche automatizzate - Fase di Consolidamento dell'architettura	100
18	Resoconto verifiche automatizzate - Fase di Realizzazione del prodotto	102
19	Metriche principali	103

1 Introduzione

1.1 Scopo del documento

Il documento ha lo scopo di definire gli obiettivi di qualità, di processo e di prodotto, e le strategie che verranno adottate dal *team_G* al fine di perseguire tali obiettivi.

Per ciascun obiettivo di qualità, inoltre, verranno individuate delle metriche sulle quali il *team_G* si baserà per misurare il livello di qualità raggiunto, sia per quanto riguarda i processi adottati, che per il prodotto sviluppato.

1.2 Scopo del prodotto

Lo scopo del prodotto è di permettere la creazione e l'esecuzione di presentazioni a partire da *mappe mentali_G*. L'utente sarà guidato nella creazione di una *mappa mentale_G* e di uno o più *percorsi di presentazione_G*, utilizzando i nodi di tale mappa. L'utente potrà eseguire una presentazione seguendo un *percorso* creato oppure visitando qualsiasi nodo della *mappa* costruita; rompendo così la sequenzialità nella presentazione. Il prodotto sarà utilizzabile attraverso un *browser_G*.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario v2.0.0*. Ogni occorrenza dei vocaboli presenti nel *Glossario* è marcata da una "G" maiuscola in pedice ed è scritta in corsivo (es: *Esempio_G*).

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v3.0.0* ;
- **Capitolato:** <http://www.math.unipd.it/~tullio/IS-1/2014/Progetto/C4.pdf>.

1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v3.0.0* ;
- **Capacity Maturity Model:** http://en.wikipedia.org/wiki/Capability_Maturity_Model;
- **Capacity Maturity Model Integration:** http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration;
- **PDCA (Plan-Do-Check-Act):** <http://it.wikipedia.org/wiki/PDCA>;
- **Standard ISO/IEC 12207:2008 - IEEE Std 12207-2008:** <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4475822>
- **Standard ISO/IEC 9126:** http://it.wikipedia.org/wiki/ISO/IEC_9126;
- **Indice di Gulpease:** http://it.wikipedia.org/wiki/Indice_Gulpease;
- **MI and MINC - Maintainability Index:** <http://www.virtualmachinery.com/sidebar4.htm>.

2 Qualità di processo

Al fine di garantire degli standard di elevata qualità nello sviluppo del software Premi, il focus è stato posto in primo luogo sui processi adottati durante tutto il ciclo di vita del prodotto. In particolare, il *team_G* ha individuato la necessità di fissare obiettivi, metriche e strategie per i processi più importanti (tratti da ISO/IEC 12207:2008) ed adottare una strategia di misurazione costante della loro qualità, tale da ottenere un tracciamento continuo del soddisfacimento degli obiettivi fissati. A questo scopo sono stati valutati vari modelli e standard per la misurazione della qualità dei processi, come CMMI e ISO/IEC 15504, ma si è deciso in fine di adottare il modello *CMM_G* (*Capability Maturity Model*) in quanto, basandosi su un minor numero di parametri, permette una valutazione più semplice ed adatta ad un *team_G* con poca esperienza come Pragma. Questo modello opera una misurazione su due parametri:

- **Capability:** caratteristica valutata sui processi presi singolarmente; determina quanto è adeguato un processo, in termini di efficienza ed efficacia, per gli scopi per i quali è stato definito;
- **Maturity:** caratteristica posseduta da un insieme di processi, risultante dalla combinazione delle capability dei processi coinvolti; essa indica quanto è governato il sistema di processi di un'azienda.

Al fine di assicurare qualità nei processi adottati, il *team_G* ha stabilito le seguenti soglie di riferimento per il modello *CMM_G*:

- **Accettabilità:** 3 – 5;
- **Ottimalità:** 4 – 5.

Le misurazioni ottenute vengono utilizzate all'interno della strategia di miglioramento continuo della qualità, realizzata attraverso il metodo **PDCA** (detto anche *Ciclo di Deming*). Tale approccio si divide in 4 passaggi principali:

- **Plan:** individuazione di obiettivi e processi necessari per il raggiungimento dei risultati attesi;
- **Do:** attuazione del piano individuato al passo precedente e raccolta di dati sulla qualità ottenuta;
- **Check:** confronto fra i risultati ottenuti (fase di *Do*) ed i risultati attesi (fase di *Plan*) ed individuazione delle differenze;
- **Act:** determinazione delle cause delle differenze fra risultati ottenuti e risultati attesi; richiesta di azioni correttive per il miglioramento della qualità.

Per garantire una buona qualità di processo, il *team_G* ha individuato dallo standard ISO/IEC 12207:2008 i processi che ritiene più importanti nell'arco del ciclo di vita del prodotto e li ha istanziati individuando obiettivi e metriche coerenti con i livelli di qualità perseguiti.

2.1 Infrastructure Management Process (6.2.2)

Il processo si pone come obiettivo quello di fornire, mantenere ed aggiornare l'infrastruttura ed i servizi necessari allo svolgimento del *progetto_G* nell'arco di tutto il suo ciclo di vita. Con il termine *infrastruttura* si intendono elementi *hardware_G*, software, metodi, strumenti, tecniche e standard impiegati nello sviluppo del prodotto.

2.1.1 Obiettivi di qualità

Per tutta la durata del *progetto_G*, l'infrastruttura impiegata nello sviluppo dovrà raggiungere determinati obiettivi; in particolare:

- tutte le procedure riguardanti le attività svolte più frequentemente durante lo sviluppo del *progetto_G* saranno descritte esaustivamente nel documento *Norme di Progetto*;
- tutti i riferimenti normativi e informativi saranno completi di informazioni utili al loro reperimento;

- la *piattaforma_G PragmaDB* sarà disponibile all'uso ogniquale volta un componente del *team_G* avesse bisogno di accedere ai dati in essa contenuti;
- i dati ottenuti da *PragmaDB* saranno sempre coerenti e aggiornati;
- nell'arco dello sviluppo del *progetto_G*, *PragmaDB* si estenderà fornendo nuove funzionalità in relazione alle nuove attività intraprese; esse risulteranno implementate e funzionanti alla prima necessità di utilizzo;
- gli script bash utilizzati forniranno documenti compilati correttamente e provvisti di termini di glossario marcati opportunamente.

2.1.2 Metriche

2.1.2.1 Disponibilità *PragmaDB*

Indica la percentuale di disponibilità di utilizzo della *piattaforma_G PragmaDB* rispetto alle richieste di accesso.

- **Misurazione:** $D = \frac{A}{R} \cdot 100$, dove A corrisponde al numero di accessi avvenuti correttamente alla pagina di login della *piattaforma_G* e R il numero totale di richieste di accesso alla pagina di login inoltrate alla *piattaforma_G*;
- **Valori di ottimalità:** 90 – 100;
- **Valori di accettazione:** 80 – 100.

2.1.2.2 Tempo di correzione incoerenze in *PragmaDB*

Indica il periodo medio intercorso fra l'individuazione di un'incoerenza nella *piattaforma_G PragmaDB* da parte di un verificatore ed il suo aggiornamento da parte di un altro componente del *team_G*.

- **Misurazione:** $T = \frac{\sum_{i=1}^n C_i}{n}$ (con T espresso in *giorni*) dove C_i è il tempo intercorso fra il momento di individuazione dell'incoerenza i in *PragmaDB* e l'istante in cui tale dato viene corretto;
- **Valori di ottimalità:** 0 – 1;
- **Valori di accettazione:** 0 – 3.

2.1.2.3 Errori di individuazione termini di glossario

Misura il numero medio di termini di glossario non marcati correttamente dallo script bash.

- **Misurazione:** $E = \frac{\sum_{i=1}^n (T_i \cdot D_i)}{\sum_{i=1}^n D_i}$ dove T_i è il numero di termini di glossario non marcati correttamente nel documento i e D_i corrisponde alla dimensione del documento i (conteggiata in *parole*);
- **Valori di ottimalità:** 0;
- **Valori di accettazione:** 0 – 3.

2.1.3 Strategie

L'infrastruttura necessaria allo svolgimento del *progetto_G* dovrà essere mantenuta costantemente aggiornata; in particolare l'utilizzo delle metriche sopra indicate permetterà l'individuazione di eventuali errori all'interno degli strumenti utilizzati, la cui correzione (da effettuarsi nel minor tempo possibile) permetterà di ripristinare l'erogazione di dati corretti e coerenti.

2.2 Project Planning, Assessment & Control Process (6.3.1 - 6.3.2)

Il macro-processo (derivante dall'unione dei processi *Project Planning Process* e *Project Assessment & Control Process*) ha lo scopo di produrre dei piani di sviluppo per il *progetto_G*, comprendenti scelta del modello di ciclo di vita del prodotto, descrizioni delle attività e dei compiti da svolgere durante tutto il ciclo di sviluppo, pianificazione temporale del lavoro e dei costi da sostenere, allocazione di compiti e responsabilità, e di effettuare periodicamente delle misurazioni per rilevare lo stato del *progetto_G* rispetto alle pianificazioni prodotte.

2.2.1 Obiettivi di qualità

L'intero sviluppo del *progetto_G* dovrà seguire la pianificazione prodotta, in particolare:

- ogni attività verrà svolta da parte di colui al quale è stata assegnata, rispettando le tempistiche fissate e svolgendo tutti i compiti nei quali è stata suddivisa;
- il costo necessario allo svolgimento di una fase di *progetto_G* non dovrà eccedere quanto preventivato per tale fase.

2.2.2 Metriche

2.2.2.1 Schedule Variance

Indica se si è in linea, in anticipo o in ritardo rispetto alla schedulazione delle attività di *progetto_G* pianificate.

- **Misurazione:** $SV = BCWP - BCWS$, dove $BCWP$ sono le attività completate ad un certo momento e $BCWS$ le attività che, secondo la pianificazione, dovrebbero essere state completate a quel momento;
- **Valori di ottimalità:** ≥ 0 ;
- **Valori di accettazione:** ≥ 0 .

2.2.2.2 Budget Variance

Indica se alla data corrente si è speso di più o di meno rispetto a quanto previsto a budget nella pianificazione.

- **Misurazione:** $BV = BCWS - ACWP$, dove $BCWS$ è il costo pianificato per realizzare le attività di *progetto_G* alla data corrente e $ACWP$ è il costo effettivamente sostenuto alla data corrente;
- **Valori di ottimalità:** ≥ 0 ;
- **Valore di accettazione:** ≥ 0 .

2.2.3 Strategie

La pianificazione effettuata dovrà essere aggiornata costantemente durante tutta l'attività di *progetto_G* per essere sempre coerente con la situazione corrente.

Qualsiasi eventuale valore negativo a livello di *Schedule Variance* o *Budget Variance* rilevato in una fase di lavoro dovrà essere assolutamente compensato entro la fine dell'attività di *progetto_G*, in quanto non è assolutamente ammesso eccedere le ore di lavoro finali e il preventivo dei costi finale indicato nella pianificazione.

2.3 Risk Management Process (6.3.4)

L'obiettivo del processo è quello di identificare, analizzare, trattare e monitorare continuamente i rischi che possono insorgere durante l'intera attività di *progetto_G*.

2.3.1 Obiettivi di qualità

Il *team_G* dovrà gestire correttamente i rischi, in particolare:

- all'inizio dell'attività di *progetto_G*, verranno individuati i principali fattori di rischio riguardanti l'organizzazione delle attività;
- all'inizio di ogni fase, l'analisi dei rischi porterà all'individuazione di nuovi rischi specifici per tale fase;
- i rischi analizzati che si paleseranno saranno trattati secondo le strategie individuate in fase di individuazione e il loro impatto sarà controllato.

2.3.2 Metriche

2.3.2.1 Rischi non preventivati

Indicatore che evidenzia i rischi non preventivati.

- **Misurazione:** indice numerico che viene incrementato nel momento in cui si manifesta un rischio non individuato nell'attività di analisi dei rischi;
- **Valori di ottimalità:** 0;
- **Valori di accettazione:** $0 - 5$.

2.3.2.2 Efficienza di gestione dei rischi

Misura il tempo medio trascorso fra l'individuazione di un rischio e il momento in cui manifesta in modo problematico i suoi effetti. Tale indicatore è importante in quanto determina se l'azione di risk management interviene correttamente su un rischio analizzato per mitigarne/allontanarne gli effetti problematici.

- **Misurazione:** $E = \frac{\sum_{i=1}^n (M_i \cdot P_i)}{\sum_{i=1}^n P_i}$, (con E espresso in *giorni*) dove M_i è il tempo intercorso fra l'individuazione del rischio i e l'istante in cui manifesta in modo problematico i suoi effetti, espresso in *giorni*, e P_i corrisponde al grado di pericolosità del rischio i , valutato in scala crescente [1-5];
- **Valori di ottimalità:** ≥ 60 ;
- **Valori di accettazione:** ≥ 20 .

2.3.3 Strategie

Il livello di probabilità dei rischi analizzati dovrà sempre essere tenuto sotto controllo.

Anche se a basso livello di pericolosità, in caso il rischio si manifestasse, il *team_G* dovrà attuare le contromisure previste al fine di mitigare i suoi effetti ed evitare che la sua pericolosità aumenti.

2.4 System/Software Requirements Analysis Process (6.4.2 - 7.1.2)

Il processo punta a trasformare i requisiti definiti dagli *stakeholder_G* in un set di requisiti tecnici che fungerà da linea guida nella progettazione del sistema.

2.4.1 Obiettivi di qualità

I requisiti identificati dal $team_G$ dovranno essere gestiti in maniera tale da raggiungere i seguenti traguardi:

- per ogni requisito verrà tenuta traccia della fonte da cui è stato ricavato;
- per ogni requisito dovrà essere possibile indicare dei test, da effettuare per verificarne il soddisfacimento da parte del prodotto;
- per ogni requisito sarà possibile ricostruire i cambiamenti principali effettuati nella sua formulazione, durante tutto il ciclo di sviluppo del prodotto;
- nessun requisito dovrà risultare superfluo o ambiguo agli $stakeholder_G$;
- tutti i requisiti che il prodotto andrà a soddisfare saranno stati precedentemente approvati dai committenti.

2.4.2 Metriche

2.4.2.1 Requisiti obbligatori soddisfatti

Indica la percentuale dei requisiti obbligatori soddisfatti dal prodotto.

- **Misurazione:** $S = \frac{N_S}{N_O} \cdot 100$, dove N_S è il numero dei requisiti obbligatori soddisfatti dal sistema e N_O è il numero dei requisiti obbligatori identificati;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 100.

2.4.2.2 Requisiti desiderabili/facoltativi accettati soddisfatti

Indica la percentuale dei requisiti desiderabili/facoltativi accettati soddisfatti dal prodotto.

- **Misurazione:** $S = \frac{N_{DAS} + N_{FAS}}{N_{DA} + N_{FA}} \cdot 100$, dove N_{DAS} è il numero dei requisiti desiderabili accettati soddisfatti dal sistema, N_{FAS} è il numero dei requisiti facoltativi accettati soddisfatti dal sistema, N_{DA} è il numero dei requisiti desiderabili accettati e N_{FA} è il numero dei requisiti facoltativi accettati;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 100.

2.4.2.3 Requisiti desiderabili/facoltativi non accettati soddisfatti

Indica la percentuale dei requisiti desiderabili/facoltativi non accettati soddisfatti dal prodotto.

- **Misurazione:** $S = \frac{N_{DNS} + N_{FNS}}{N_{DN} + N_{FN}} \cdot 100$, dove N_{DNS} è il numero dei requisiti desiderabili non accettati soddisfatti dal sistema, N_{FNS} è il numero dei requisiti facoltativi non accettati soddisfatti dal sistema, N_{DN} è il numero dei requisiti desiderabili non accettati e N_{FN} è il numero dei requisiti facoltativi non accettati;
- **Valori di ottimalità:** 50 – 100;
- **Valori di accettazione:** 0 – 100.

2.4.3 Strategie

Tutti i requisiti individuati dovranno essere correttamente inseriti nella $piattaforma_G$ *PragmaDB*, la quale si occuperà di mantenere traccia delle fonti dalle quali derivano, delle modifiche effettuate e della loro implementazione nel prodotto.

2.5 System/Software Architectural Design Process (6.4.3 - 7.1.3)

Il processo si pone come obiettivo quello di identificare una corrispondenza fra requisiti di sistema ed elementi del sistema.

2.5.1 Obiettivi di qualità

Durante lo svolgimento delle attività previste da questo processo, il *team_G* punterà a definire un'architettura adatta agli scopi del *progetto_G*:

- ogni componente progettato come parte del sistema risulterà essere necessario per il funzionamento del prodotto e, quindi, costantemente tracciabile ai requisiti che soddisfa;
- il sistema dovrà presentare basso accoppiamento ed alta coesione
- ogni componente dovrà essere progettato puntando su incapsulamento, modularizzazione e riuso di codice.

2.5.2 Metriche

2.5.2.1 Structural Fan-In

In riferimento ad un modulo del software, misura quanti altri moduli lo utilizzano durante la loro esecuzione; tale indicazione permette di stabilire il livello di riuso implementato.

- **Misurazione:** indice numerico che incrementa nel momento in cui viene individuato un modulo che, durante la sua esecuzione, chiama il modulo in oggetto;
- **Valori di ottimalità:** ≥ 2 ;
- **Valori di accettazione:** ≥ 0 .

2.5.2.2 SFIN - Ottimalità

In riferimento alla metrica *Structural Fan-In*, indica la percentuale di moduli che rispettano i parametri di ottimalità per essa definiti; questa indicazione permette di monitorare il numero di moduli ad alto livello di riuso.

- **Misurazione:** $SFIN_O = \frac{N_{MO}}{N_M} \cdot 100$, dove N_{MO} è il numero di moduli che rispettano i parametri di ottimalità per la metrica *Structural Fan-In* e N_M è il numero totale di moduli definiti nell'architettura;
- **Valori di ottimalità:** ≥ 50 ;
- **Valori di accettazione:** ≥ 30 .

2.5.2.3 Structural Fan-Out

In riferimento ad un modulo del software, misura quanti moduli vengono utilizzati durante la sua esecuzione; tale indicazione permette di stabilire il livello di accoppiamento implementato.

- **Misurazione:** indice numerico che incrementa nel momento in cui viene individuato un modulo utilizzato dal modulo in oggetto durante la sua esecuzione;
- **Valori di ottimalità:** $0 - 1$;
- **Valori di accettazione:** $0 - 5$.

2.5.2.4 SFOUT - Non Accettabilità

In riferimento alla metrica *Structural Fan-Out*, indica la percentuale di moduli che non rispettano i parametri di accettazione per essa definiti; questa indicazione permette di monitorare la presenza di un numero eccessivo di moduli ad alto livello di accoppiamento.

- **Misurazione:** $SFOUT_{NA} = \frac{N_{MNA}}{N_M} \cdot 100$, dove N_{MNA} è il numero di moduli che non rispettano i parametri di accettazione per la metrica *Structural Fan-Out* e N_M è il numero totale di moduli definiti nell'architettura;
- **Valori di ottimalità:** 0 – 3;
- **Valori di accettazione:** 0 – 6.

2.5.3 Strategie

Nel corso dell'attività di progettazione (sia ad alto livello che di dettaglio) le componenti verranno inserite nella *piattaforma_G PragmaDB*, la quale si occuperà di mantenere aggiornati i tracciamenti fra esse ed i requisiti che soddisfano, oltre alle relazioni presenti fra le varie componenti.

2.6 Software Detailed Design Process (7.1.4)

Lo scopo del processo è fornire una progettazione di dettaglio del prodotto che andrà ad implementare i requisiti individuati.

2.6.1 Obiettivi di qualità

Le attività svolte dovranno raggiungere i seguenti obiettivi:

- il livello di dettaglio della progettazione dovrà essere tale da guidare codifica e testing senza bisogno di informazioni aggiuntive, indicando metodi (corredati da parametri) e campi dati forniti da ciascuna classe;
- la struttura a basso livello dell'architettura e le relazioni fra le varie unità software concepite saranno espone chiaramente nel documento di *Definizione di Prodotto*, che definirà dettagliatamente cosa implementare;
- oltre alle unità software individuate, le attività permetteranno di definire dettagliatamente le interfacce fra esse costituite.

2.6.2 Metriche

2.6.2.1 Numero di metodi per classe

Indica il numero di metodi definiti in una classe; un valore molto alto potrebbe indicare una cattiva decomposizione delle funzionalità a livello di progettazione.

- **Misurazione:** indice numerico che indica il numero di metodi definiti in una classe;
- **Valori di ottimalità:** 1 – 7;
- **Valori di accettazione:** 1 – 10.

2.6.2.2 Metodi per classe - Non Accettabilità

In riferimento alla metrica *Numero di metodi per classe*, indica la percentuale di classi che non rispettano i parametri di accettazione per essa definiti; questa indicazione permette di monitorare il livello di decomposizione delle funzionalità raggiunto.

- **Misurazione:** $C_{NA} = \frac{N_{CNA}}{N_C} \cdot 100$, dove N_{CNA} è il numero di classi che non rispettano i parametri di accettazione per la metrica *Numero di metodi per classe* e N_C è il numero totale di classi definite nell'architettura;
- **Valori di ottimalità:** 0 – 5;
- **Valori di accettazione:** 0 – 15.

2.6.2.3 Numero di parametri per metodo

Indica il numero di parametri passati ad un metodo; un valore molto alto potrebbe indicare un metodo troppo complesso e non efficacemente suddiviso in sotto-metodi.

- **Misurazione:** indice numerico che indica il numero di parametri passato ad un metodo;
- **Valori di ottimalità:** 0 – 4;
- **Valori di accettazione:** 0 – 8.

2.6.2.4 Parametri per metodo - Non Accettabilità

In riferimento alla metrica *Numero di parametri per metodo*, indica la percentuale di metodi che non rispettano i parametri di accettazione per essa definiti; questa indicazione permette di monitorare il livello di complessità dei metodi definiti nell'architettura.

- **Misurazione:** $M_{NA} = \frac{N_{MNA}}{N_M} \cdot 100$, dove N_{MNA} è il numero di metodi che non rispettano i parametri di accettazione per la metrica *Numero di parametri per metodo* e N_M è il numero totale di metodi definiti nell'architettura;
- **Valori di ottimalità:** 0 – 3;
- **Valori di accettazione:** 0 – 5.

2.6.3 Strategie

Sarà necessario effettuare un'analisi dettagliata delle componenti individuate in progettazione architeturale, suddividendole in unità che siano facilmente codificabili e testabili per le attività successive.

2.7 Software Construction Process (7.1.5)

Il processo definisce le attività principali volte alla produzione di unità software eseguibili che riflettano quanto identificato a livello di progettazione.

2.7.1 Obiettivi di qualità

Le unità software prodotte dovranno risultare di qualità; a questo fine il *team_G* si è posto i seguenti obiettivi:

- l'implementazione delle classi e dei metodi definiti in progettazione dovrà puntare a produrre codice a bassa complessità, in modo tale che quanto prodotto risulti facilmente comprensibile e testabile;
- l'uso di costrutti e tecniche che creano sdoppiamenti del flusso di esecuzione verrà valutato attentamente ed attuato solo se strettamente necessario;

- il codice prodotto dovrà risultare facilmente manutenibile;
- il codice prodotto risulterà privo di elementi inutilizzati.

2.7.2 Metriche

2.7.2.1 Produttività di codifica

Indica il numero medio di linee di codice prodotto per ora/persona.

- **Misurazione:** $P = \frac{N_{SLOC}}{h_P}$, dove N_{SLOC} è il numero di *Source Lines Of Code* prodotte e h_P è il numero di ore/persona utilizzate per la codifica;
- **Valori di ottimalità:** ≥ 10 ;
- **Valori di accettazione:** ≥ 3 .

2.7.2.2 Complessità Ciclomatica

Indica la complessità di un programma misurando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso.

- **Misurazione:** indice numerico che indica il numero cammini percorribili nel grafo di controllo di flusso di un metodo;
- **Valori di ottimalità:** $0 - 4$;
- **Valori di accettazione:** $0 - 8$.

2.7.2.3 Complessità Ciclomatica - Non Accettabilità

In riferimento alla metrica *Complessità Ciclomatica*, indica la percentuale di unità software che non rispettano i parametri di accettazione per essa definiti; questa indicazione permette di monitorare il livello di complessità delle unità definite nell'architettura.

- **Misurazione:** $C_{NA} = \frac{N_{UNA}}{N_U} \cdot 100$, dove N_{UNA} è il numero di unità che non rispettano i parametri di accettazione per la metrica *Complessità Ciclomatica* e N_U è il numero totale di unità definite nell'architettura;
- **Valori di ottimalità:** 0 ;
- **Valori di accettazione:** $0 - 1$.

2.7.2.4 Linee di commento su linee di codice

Indica la percentuale di linee di commento presenti all'interno del codice sorgente; la loro presenza permette una più semplice comprensione ed un maggior livello di manutenibilità di quanto prodotto.

- **Misurazione:** $P = \frac{N_C}{N_{SLOC}} \cdot 100$, dove N_C è il numero di linee di commento presenti nel codice e N_{SLOC} è il numero di *Source Lines Of Code* prodotte;
- **Valori di ottimalità:** ≥ 30 ;
- **Valori di accettazione:** ≥ 25 .

2.7.2.5 Chiamate annidate

Indica il numero di funzioni o procedure chiamate all'interno di un metodo; un numero eccessivo potrebbe causare dei problemi a livello di *stack_G*.

- **Misurazione:** indice numerico che indica il numero di chiamate a funzioni o procedure presenti all'interno di un metodo;
- **Valori di ottimalità:** 0 – 5;
- **Valori di accettazione:** 0 – 8.

2.7.2.6 Variabili inutilizzate

Indica la percentuale di variabili dichiarate che non vengono mai utilizzate durante l'esecuzione.

- **Misurazione:** $P = \frac{N_{VI}}{N_{VD}} \cdot 100$, dove N_{VI} è il numero di variabili che non vengono mai utilizzate e N_{VD} è il numero di variabili dichiarate;
- **Valori di ottimalità:** 0;
- **Valori di accettazione:** 0.

2.7.2.7 Dipendenze

Misura il numero medio di chiamate *require* di una funzione, analizzate staticamente considerando la firma del metodo.

- **Misurazione:** $D = \frac{\sum_{i=1}^n R_i}{n}$, dove R_i è il numero di chiamate *require* effettuate dalla funzione i ;
- **Valori di ottimalità:** 0 – 5;
- **Valori di accettazione:** 0 – 10.

2.7.2.8 Halstead Difficulty per-function

Misura il livello di complessità di una funzione.

- **Misurazione:** $DIF = \frac{UOP}{2} \cdot \frac{OD}{UOD}$, dove UOP è il numero di operatori distinti, OD è il numero totale di operandi e UOD è il numero di operandi distinti;
- **Valori di ottimalità:** 0 – 15;
- **Valori di accettazione:** 0 – 25.

2.7.2.9 Halstead Difficulty per-function - Non Accettabilità

In riferimento alla metrica *Halstead Difficulty per-function*, indica la percentuale di unità software che non rispettano i parametri di accettabilità per essa definiti; questa indicazione permette di monitorare il livello di complessità delle unità definite nell'architettura.

- **Misurazione:** $HD_{NA} = \frac{N_{UNA}}{N_U} \cdot 100$, dove N_{UNA} è il numero di unità che non rispettano i parametri di accettabilità per la metrica *Halstead Difficulty per-function* e N_U è il numero totale di unità definite nell'architettura;
- **Valori di ottimalità:** 0;
- **Valori di accettazione:** 0 – 1.

2.7.2.10 Halstead Volume per-function

Indica la dimensione dell'implementazione di un algoritmo; si basa sul numero di operazioni eseguite e sugli operandi di una funzione.

- **Misurazione:** $VOL = (OP + OD) \cdot \log_2(UOP + UOD)$, dove OP è il numero totale di operatori, OD è il numero totale di operandi, UOP è il numero di operatori distinti e UOD è il numero di operandi distinti;
- **Valori di ottimalità:** 20 – 1000;
- **Valori di accettazione:** 20 – 1500.

2.7.2.11 Halstead Effort per-function

Rappresenta il costo necessario a scrivere il codice di una funzione.

- **Misurazione:** $E = DIF \cdot VOL$, dove DIF indica l'*Halstead Difficulty* e VOL è l'*Halstead Volume*;
- **Valori di ottimalità:** 0 – 300;
- **Valori di accettazione:** 0 – 400.

2.7.2.12 Indice di manutenibilità

Permette di stabilire quanto sarà semplice mantenere il codice prodotto.

- **Misurazione:** $MI = 171 - 3.42 \cdot \ln(aveE) - 0.23 \cdot \ln(aveV) - 16.2 \cdot \ln(aveLOC)$, dove $aveE$ è l'*Halstead Effort* medio per modulo, $aveV$ è la *complessità ciclomatica* media per modulo, e $aveLOC$ è il numero medio di linee di codice per modulo;
- **Valori di ottimalità:** 120 – 171;
- **Valori di accettazione:** 100 – 171.

2.7.2.13 Indice di manutenibilità - Non Accettabilità

In riferimento alla metrica *Indice di manutenibilità*, indica la percentuale di unità software che non rispettano i parametri di accettabilità per essa definiti; questa indicazione permette di monitorare il livello di complessità delle unità definite nell'architettura.

- **Misurazione:** $MI_{NA} = \frac{N_{UNA}}{N_U} \cdot 100$, dove N_{UNA} è il numero di unità che non rispettano i parametri di accettabilità per la metrica *Indice di manutenibilità* e N_U è il numero totale di unità definite nell'architettura;
- **Valori di ottimalità:** 0 – 5;
- **Valori di accettazione:** 0 – 10.

2.7.3 Strategie

Durante l'attività di codifica, il *Programmatore* dovrà attenersi a quanto indicato nel documento *Definizione di Prodotto*, concentrandosi (in particolare) nel limitare la complessità del codice prodotto. Sarà necessario inoltre procedere con la codifica dei test individuati nell'attività di progettazione, in modo tale da consentire la verifica del corretto funzionamento delle varie unità prodotte.

2.8 System/Software Integration Process (6.4.5 - 7.1.6)

Il processo si occupa di integrare fra loro gli elementi del sistema, rispettando quanto stabilito nell'attività di progettazione, al fine di produrre un prodotto completo tale da soddisfare quanto espresso dai requisiti identificati.

2.8.1 Obiettivi di qualità

Le attività previste da questo processo dovranno puntare a raggiungere un alto livello di automazione, in particolare:

- l'integrazione delle varie parti del sistema sarà completamente automatizzata utilizzando lo strumento di *continuous integration_G Jenkins*;
- il livello di integrazione raggiunto del sistema sarà sempre consultabile grazie all'utilizzo dello strumento di *continuous integration_G Jenkins*.

2.8.2 Metriche

2.8.2.1 Componenti integrate

Indica la percentuale di componenti progettate, attualmente implementate e correttamente integrate nel sistema.

- **Misurazione:** $I = \frac{N_{CI}}{N_{CP}} \cdot 100$, dove N_{CI} è il numero di componenti attualmente integrate nel sistema e N_{CP} è il numero di componenti delineate nell'attività di progettazione;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 100.

2.8.3 Strategie

Sarà necessario configurare accuratamente lo strumento di *continuous integration_G Jenkins* affinché esegua dei test di integrazione di quanto prodotto prima che le ultime modifiche diventino parte del sistema.

2.9 System/Software Qualification Testing Process (6.4.6 - 7.1.7)

Lo scopo del processo è quello di assicurare che ogni requisito individuato sia stato implementato nel prodotto.

2.9.1 Obiettivi di qualità

Durante lo svolgimento delle attività, ci si impegnerà affinché:

- le attività di test previste dal processo verranno svolte su un sistema le cui componenti sono verificate e correttamente integrate fra loro;
- il sistema dovrà implementare tutti i requisiti obbligatori individuati nell'attività di analisi.

2.9.2 Metriche

2.9.2.1 Test di sistema superati

Indica la percentuale di test di sistema superati.

- **Misurazione:** $S = \frac{N_{TS}}{N_{TE}} \cdot 100$, dove N_{TS} è il numero di test di sistema superati e N_{TE} è il numero di test di sistema eseguiti;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 100.

2.9.3 Strategie

Bisognerà cercare di implementare il maggior livello possibile di automazione nell'esecuzione dei test di sistema, in modo tale che la loro esecuzione non richieda costi eccessivi (soprattutto in termini temporali) e sia possibile eseguirne un numero sufficiente a garantire un'ottima copertura dei requisiti; a tal fine molto importante sarà lo strumento di *continuous integration*_G *Jenkins*, il quale andrà opportunamente configurato per eseguire i test stabiliti.

2.10 Software Documentation Management Process (7.2.1)

Il processo punta a produrre e mantenere le informazioni sul software prodotte dai processi attuati.

2.10.1 Obiettivi di qualità

Il processo di documentazione dovrà perseguire le seguenti direttive:

- la documentazione prodotta dovrà essere chiara e comprensibile a tutti gli *stakeholder*_G e sarà resa disponibile alle parti interessate per la consultazione;
- ogni forma di ambiguità sul significato di un termine utilizzato verrà eliminata grazie al *Glossario*;
- la documentazione prodotta sarà sempre aggiornata ed allineata allo stato attuale del processo di sviluppo del prodotto.

2.10.2 Metriche

2.10.2.1 Indice Gulpease

Misura la leggibilità e la complessità di un documento.

- **Misurazione:** $G = 89 + \frac{300 \cdot N_F - 10 \cdot N_L}{N_P}$, dove N_F è il numero di frasi, N_L è il numero di lettere e N_P è il numero di parole presenti nel testo;
- **Valori di ottimalità:** 50 – 100;
- **Valori di accettazione:** 40 – 100.

2.10.3 Strategie

Durante la stesura della documentazione, ogni termine con significato ambiguo deve essere indicato (corredato di definizione) nel *Glossario*; gli script automatici provvederanno alla sua segnalazione all'interno del documento.

Ogni documento sarà dotato di numero di versione e corredato da un diario delle modifiche che consente di prendere visione di tutte le azioni effettuate sul testo in oggetto.

2.11 Software Verification Process (7.2.4)

Il processo punta a verificare se qualsiasi elemento del sistema (software e non) soddisfa completamente i requisiti ad esso correlati.

2.11.1 Obiettivi di qualità

Al fine di garantire qualità nell'attuazione del processo:

- la documentazione verrà verificata attraverso *inspection*_G, poiché (effettuando un'analisi mirata degli errori) permette risparmio in termini di tempi e costi;
- i test dinamici effettuati sui vari elementi saranno il più possibile automatizzabili;
- i test dinamici effettuati sui vari elementi del software copriranno una grande parte delle possibili casistiche d'utilizzo.

2.11.2 Metriche

2.11.2.1 Copertura dei test

Indica la percentuale di *branch coverage* rilevata sul codice, ossia la percentuale di rami decisionali percorsi dalla suite di test utilizzata.

- **Misurazione:** $C = \frac{R_P}{R_T} \cdot 100$, dove R_P è il numero dei rami decisionali percorsi dai test e R_T è il numero di rami decisionali definiti nel software;
- **Valori di ottimalità:** 80 – 100;
- **Valori di accettazione:** 70 – 100.

2.11.3 Strategie

Durante le attività di correzione della documentazione, gli errori più frequenti rilevati saranno riportati in un documento, in modo tale da diventare uno dei punti chiave delle successive attività di inspection_G.

Per ogni test effettuato verrà tenuto tracciamento del suo esito.

3 Qualità di prodotto

Per garantire una buona qualità di prodotto, il $team_G$ ha individuato dallo standard ISO/IEC 9126 le qualità che ritiene più importanti nell'arco del ciclo di vita del prodotto e le ha istanziate individuando obiettivi e metriche coerenti con i livelli di qualità perseguiti.

3.1 Funzionalità (6.1)

La *funzionalità* rappresenta la capacità del software prodotto di fornire le funzioni richieste da parte del *Proponente_G*.

3.1.1 Obiettivi di qualità

Il $team_G$ si impegnerà affinché:

- **Appropriatezza (6.1.1):** tutte le funzionalità fornite dal prodotto saranno coerenti con le aspettative del cliente;
- **Accuratezza (6.1.2):** il prodotto fornirà i risultati attesi, con il livello di dettaglio richiesto;
- **Sicurezza (6.1.4):** il prodotto si occuperà di proteggere le informazioni e i dati in modo tale che persone o sistemi non autorizzati non possano accedervi per leggerli o modificarli.

3.1.2 Metriche

3.1.2.1 Completezza dell'implementazione funzionale

Indica la percentuale di requisiti funzionali coperti dall'implementazione.

- **Misurazione:** $C = (1 - \frac{N_{FM}}{N_{FI}}) \cdot 100$, dove N_{FM} è il numero di funzionalità mancanti nell'implementazione e N_{FI} è il numero di funzionalità individuate nell'attività di analisi;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 100.

3.1.2.2 Accuratezza rispetto alle attese

Indica la percentuale di risultati concordi alle attese.

- **Misurazione:** $A = (1 - \frac{N_{RD}}{N_{TE}}) \cdot 100$, dove N_{RD} è il numero di test che producono risultati discordanti rispetto alle attese e N_{TE} è il numero di test-case eseguiti;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** $90 - 100$.

3.1.2.3 Controllo degli accessi

Indica la percentuale di operazioni illegali non bloccate.

- **Misurazione:** $I = \frac{N_{IE}}{N_{II}} \cdot 100$, dove N_{IE} è il numero di operazioni illegali effettuabili dai test e N_{II} è il numero di operazioni illegali individuate;
- **Valori di ottimalità:** 0;
- **Valori di accettazione:** $0 - 10$.

3.2 Affidabilità (6.2)

L'*affidabilità* rappresenta la capacità del prodotto software di svolgere correttamente le sue funzioni quando viene utilizzato sotto determinate condizioni.

3.2.1 Obiettivi di qualità

L'esecuzione del prodotto dovrà presentare le seguenti caratteristiche:

- **Maturità (6.2.1):** il software impedirà l'esecuzione di malfunzionamenti, operazioni illegali e restituzione di risultati errati (*failure*) in seguito a *fault*;
- **Fault tolerance (6.2.2):** il prodotto sarà in grado di gestire situazioni di errore anche in seguito all'insorgere di guasti.

3.2.2 Metriche

3.2.2.1 Densità di failure

Indica la percentuale di operazioni di testing conclusesi in failure.

- **Misurazione:** $F = \frac{N_{FE}}{N_{TE}} \cdot 100$, dove N_{FE} è il numero di failure rilevati durante l'attività di testing e N_{TE} è il numero di test-case eseguiti;
- **Valori di ottimalità:** 0;
- **Valori di accettazione:** 0 – 10.

3.2.2.2 Blocco di operazioni non corrette

Indica la percentuale di funzionalità in grado di gestire correttamente i fault che potrebbero verificarsi.

- **Misurazione:** $B = \frac{N_{FE}}{N_{ON}} \cdot 100$, dove N_{FE} è il numero di failure evitati durante i test effettuati e N_{ON} è il numero di test-case eseguiti che prevedono l'esecuzione di operazioni non corrette, causa di possibili failure;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 80 – 100.

3.3 Usabilità (6.3)

L'*usabilità* rappresenta la capacità del prodotto di essere comprensibile, apprendibile, e attraente per l'utente.

3.3.1 Obiettivi di qualità

Il prodotto dovrà puntare ai seguenti obiettivi di usabilità:

- **Comprensibilità (6.3.1):** il software permetterà all'utente di riconoscerne le funzionalità e comprenderne le modalità di utilizzo per raggiungere i risultati attesi;
- **Apprendibilità (6.3.2):** il prodotto semplificherà all'utente l'apprendimento del suo utilizzo;
- **Operabilità (6.3.3):** il prodotto metterà a proprio agio l'utente presentandogli messaggi e funzionalità coerenti con le sue aspettative;
- **Attrattiva (6.3.4):** il prodotto presenterà delle funzionalità e delle interfacce utente che lo renderanno attrattivo per l'utilizzatore.

3.3.2 Metriche

3.3.2.1 Comprensibilità delle funzioni offerte

Indica la percentuale di operazioni comprese in modo immediato dall'utente, senza la consultazione del manuale.

- **Misurazione:** $C = \frac{N_{FC}}{N_{FO}} \cdot 100$, dove N_{FC} è il numero di funzionalità comprese in modo immediato dall'utente durante l'attività di testing del prodotto e N_{FO} è il numero di funzionalità offerte dal sistema;
- **Valori di ottimalità:** 90 – 100;
- **Valori di accettazione:** 80 – 100.

3.3.2.2 Facilità di apprendimento delle funzionalità

Indica il tempo medio impiegato dall'utente nell'imparare ad usare correttamente una data funzionalità.

- **Misurazione:** indicatore numerico (espresso in *minuti*) che tiene traccia del tempo medio impiegato dall'utente nell'apprendere il corretto utilizzo di una funzionalità offerta dal sistema;
- **Valori di ottimalità:** 0 – 15;
- **Valori di accettazione:** 0 – 30.

3.3.2.3 Consistenza operativa in uso

Indica la percentuale di messaggi e funzionalità offerte all'utente che rispettano le sue aspettative riguardo al comportamento del software.

- **Misurazione:** $C = (1 - \frac{N_{MFI}}{N_{MFO}}) \cdot 100$, dove N_{MFI} è il numero di messaggi e funzionalità che non rispettano le aspettative dell'utente e N_{MFO} è il numero di messaggi e funzionalità offerti dal sistema;
- **Valori di ottimalità:** 90 – 100;
- **Valori di accettazione:** 80 – 100.

3.3.2.4 Elementi personalizzabili

Indica la percentuale di elementi personalizzabili rispetto a quanto l'utente vorrebbe personalizzare.

- **Misurazione:** $C = \frac{N_{EC}}{N_{PD}} \cdot 100$, dove N_{EC} è il numero di elementi per i quali è stata prevista personalizzazione da parte dell'utente e N_{PD} è il numero di elementi che l'utente desidererebbe personalizzare;
- **Valori di ottimalità:** 90 – 100;
- **Valori di accettazione:** 60 – 100.

3.4 Efficienza (6.4)

L'*efficienza* rappresenta la capacità del prodotto software di fornire le funzionalità offerte con performance appropriate, relativamente a risorse usate, sotto determinate condizioni.

3.4.1 Obiettivi di qualità

Il prodotto dovrà essere efficiente, in particolare:

- **Efficienza nel tempo (6.4.1):** il software dovrà fornire appropriati tempi di risposta e di elaborazione nell'eseguire le funzionalità previste sotto determinate condizioni di utilizzo;
- **Utilizzo delle risorse (6.4.2):** il prodotto dovrà rispettare dei limiti in termini di utilizzo di risorse, per evitare che $client_G$ e $server_G$ siano sottoposti a stress di calcolo, o debbano attendere a lungo se vi è uno scambio eccessivo di dati attraverso la connessione internet.

3.4.2 Metriche

3.4.2.1 Tempo di risposta

Indica il periodo temporale medio che intercorre fra la richiesta al software di una determinata funzionalità e la restituzione del risultato all'utente.

- **Misurazione:** $T_{RISP} = \frac{\sum_{i=1}^n T_i}{n}$ (con T_{RISP} espresso in *secondi*) dove T_i è il tempo intercorso fra la richiesta i di una funzionalità ed il completamento delle operazioni necessarie a restituire un risultato a tale richiesta;
- **Valori di ottimalità:** 0 – 3;
- **Valori di accettazione:** 0 – 8.

3.5 Manutenibilità (6.5)

La *manutenibilità* rappresenta la capacità del prodotto di essere modificato, tramite correzioni, miglioramenti o adattamenti del software a modifiche negli ambienti, nei requisiti e nelle specifiche funzionali.

3.5.1 Obiettivi di qualità

Le operazioni di manutenzione andranno agevolate il più possibile adottando le seguenti caratteristiche:

- **Analizzabilità (6.5.1):** il software dovrà consentire a coloro che si occuperanno di mantenerlo di identificare velocemente le possibili cause di errori e malfunzionamenti;
- **Modificabilità (6.5.2):** gli obiettivi di qualità fissati per le attività di progettazione e di codifica permetteranno ai manutentori di applicare cambiamenti a componenti del prodotto;
- **Stabilità (6.5.3):** gli obiettivi di qualità fissati per le attività di progettazione e di codifica permetteranno ai manutentori di evitare l'insorgenza di effetti indesiderati a seguito di modifiche effettuate ad alcune componenti del prodotto;
- **Testabilità (6.5.4):** gli obiettivi di qualità fissati per le attività di progettazione e di codifica permetteranno ai manutentori di verificare e validare attraverso test i cambiamenti effettuati a componenti del prodotto.

3.5.2 Metriche

3.5.2.1 Capacità di analisi di failure

Indica la percentuale di failure registrate, delle quali sono state individuate le cause.

- **Misurazione:** $I = \frac{N_{FI}}{N_{FR}} \cdot 100$, dove N_{FI} è il numero di failure delle quali sono state individuate le cause e N_{FR} è il numero di failure rilevate;

- **Valori di ottimalità:** 80 – 100;
- **Valori di accettazione:** 60 – 100.

3.5.2.2 Impatto delle modifiche

Indica la percentuale di modifiche effettuate in risposta a failure che hanno portato all'introduzione di nuove failure in altre componenti del sistema.

- **Misurazione:** $I = \frac{N_{FRE}}{N_{FR}} \cdot 100$, dove N_{FRE} è il numero di failure risolte con l'introduzione di nuove failure e N_{FR} è il numero di failure risolte;
- **Valori di ottimalità:** 0 – 10;
- **Valori di accettazione:** 0 – 20.

3.6 Portabilità (6.6)

La *portabilità* rappresenta la capacità del software di poter essere trasportato da un ambiente ad un altro.

3.6.1 Obiettivi di qualità

Sarà agevolata la portabilità del prodotto adottando i seguenti obiettivi:

- **Adattabilità (6.6.1):** il software sarà in grado di adattare le sue funzionalità a tutti gli ambienti individuati nell'analisi dei requisiti;
- **Sostituibilità (6.6.4):** il software permetterà all'utente di mantenere lo stesso metodo di lavoro e la stessa visione ad alto livello adottata nell'utilizzo di un prodotto precedentemente utilizzato per i medesimi scopi e nello stesso ambiente.

3.6.2 Metriche

3.6.2.1 Versioni dei browser supportate

Indica la percentuale di versioni di $browser_G$ attualmente supportate, fra quelle individuate dai requisiti.

- **Misurazione:** $S = \frac{N_{VS}}{N_{VI}} \cdot 100$, dove N_{VS} è il numero di versioni di $browser_G$ supportate dal prodotto e N_{VI} è il numero di versioni di $browser_G$ che devono essere supportate dal prodotto;
- **Valori di ottimalità:** 100;
- **Valori di accettazione:** 100.

3.6.2.2 Inclusione di funzionalità da altri prodotti

Indica la percentuale di funzionalità del software utilizzato in precedenza dall'utente che produce risultati simili a quelli ottenuti dal prodotto in oggetto.

- **Misurazione:** $I = \frac{N_{FPA}}{N_{FPP}} \cdot 100$, dove N_{FPA} è il numero di funzionalità del software utilizzato in precedenza dall'utente che produce risultati simili a quelli ottenuti dal prodotto in oggetto e N_{FPP} è il numero di funzionalità offerte dal software utilizzato in precedenza dall'utente;
- **Valori di ottimalità:** 90 – 100;
- **Valori di accettazione:** 80 – 100.

4 Specifica dei test

Il *team_G* Pragma, al fine di implementare del software che sia di qualità, ha strutturato dei test atti a verificare che il software prodotto rispecchi le funzionalità a fronte di risultati attesi. Questi test sono il frutto dell'applicazione delle tecniche di verifica dinamica che sono state introdotte nel documento *Norme di Progetto v3.0.0*. Tutte le attività di testing prodotte devono poter essere ripetibili e devono essere deterministiche, al fine di poter fornire delle informazioni utili a intraprendere azioni di correzione, nel caso in cui i risultati ottenuti siano diversi da quelli attesi. Per avere un tracciamento dei test prodotti e dei risultati ottenuti si è scelto di classificare il tutto producendo dei log che siano di facile consultazione e che possano fornire una precisa indicazione di quelli che sono stati gli output di queste attività di verifica, eventuali errori o eventuali risultati che siano non coerenti con quanto in precedenza fissato.

4.1 Tipi di test

Sono stati individuati quattro livelli di testing e sono rispettivamente:

- **Test di unità [TU]:** con questa tipologia di test si cerca di verificare la più piccola parte di lavoro prodotta da un programmatore. Questo si traduce tendenzialmente a verificare i metodi e le funzioni scritte;
- **Test di integrazione [TI]:** con questa tipologia di test si cerca di verificare le componenti di sistema, descritte in *Specifica Tecnica v1.0.0*. Più precisamente, l'obiettivo è quello di testare il funzionamento dei vari package prodotti, sia singolarmente che nel loro insieme;
- **Test di sistema [TS]:** con questa tipologia di test si cerca di verificare che il comportamento e il funzionamento dell'architettura siano corretti;
- **Test di validazione [TV]:** con questa tipologia di test si vuole verificare che il lavoro prodotto soddisfi quanto richiesto dal *proponente_G*.

4.2 Test di Validazione

I test di validazione hanno lo scopo di accertare che tutte le funzionalità richieste dal *proponente_G* siano state soddisfatte. Per questo motivo, attraverso delle macro azioni, si andrà a simulare il comportamento generale dell'applicativo e dell'utente che interagisce con esso.

I test di validazione saranno organizzati nel modo seguente:

TV[TipoRequisito][ImportanzaRequisito][IdRequisito]

dove:

- **TipoRequisito** può assumere valori tra:
 - *F* per i requisiti funzionali;
 - *Q* per i requisiti di qualità;
 - *V* per i requisiti di vincolo;
 - *P* per i requisiti prestazionali.
- **ImportanzaRequisito** può assumere valori tra:
 - *D* per i requisiti desiderabili;
 - *O* per i requisiti di obbligatori;
 - *F* per i requisiti di facoltativi.
- **IdRequisito** assume un valore gerarchico che identifica il singolo requisito.

Id Test	Descrizione	Stato
TVFO1	<p>L'utente autenticato intende creare un nuovo <i>progetto_G</i>. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>Dashboard</i>; • Premere il pulsante <i>Crea nuovo progetto_G</i>; • Inserire un nome per il nuovo <i>progetto_G</i>; • Premere il pulsante di conferma creazione nuovo <i>progetto_G</i>; • Verificare che sia stato creato un nuovo <i>progetto_G</i> con il nome indicato. 	<i>Non Implementato</i>
TVFO2	<p>L'utente autenticato intende creare un nuovo <i>progetto_G</i> da desktop. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>Dashboard</i>; • Premere il pulsante <i>Crea nuovo progetto_G</i>; • Inserire un nome per il nuovo <i>progetto_G</i>; • Premere il pulsante di conferma creazione nuovo <i>progetto_G</i>; • Verificare che sia stato creato un nuovo <i>progetto_G</i> con il nome indicato. 	<i>Non Implementato</i>
TVFO4	<p>L'utente autenticato intende modificare un <i>progetto_G</i> esistente. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>Dashboard</i>; • Premere il pulsante di modifica relativo al <i>progetto_G</i> che si desidera modificare; • Dalla <i>EditMapView</i> selezionare una delle opzioni di modifica proposte; • Verificare che la modifica scelta sia stata apportata correttamente. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD4.1	<p>L'utente autenticato intende modificare il nome del <i>progetto_G</i>. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Inserire il nuovo nome del <i>progetto_G</i>; • Premere il pulsante di conferma; • Verificare che il nome del <i>progetto_G</i> sia stato modificato correttamente. 	<i>Non Implementato</i>
TVFO4.2	<p>L'utente autenticato intende modificare la struttura della mappa mentale. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Selezionare un nodo ed effettuare una delle possibili modifiche; • Verificare che la modifica scelta sia stata effettuata. 	<i>Non Implementato</i>
TVFO4.2.3	<p>L'utente autenticato intende modificare un nodo della mappa. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Selezionare il nodo interessato alla modifica; • Effettuare la modifica al nodo scegliendo fra quelle proposte; • Verificare che le modifiche scelte siano state apportate correttamente. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD4.3	<p>L'utente autenticato intende creare un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Selezionare l'opzione <i>Aggiungi percorso_G</i>; • Inserire un nome per il nuovo <i>percorso_G</i>; • Confermare la creazione del nuovo <i>percorso_G</i>; • Verificare che il nuovo <i>percorso_G</i> sia stato creato con il nome specificato. 	<i>Non Implementato</i>
TVFD4.3.2	<p>L'utente autenticato intende scegliere un nodo di una <i>mappa mentale_G</i> come primo <i>frame_G</i> di un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Aver iniziato la procedura di creazione di un <i>percorso di presentazione_G</i> personalizzato; • Selezionare il nodo della <i>mappa mentale_G</i> che intende aggiungere al <i>percorso_G</i>; • Premere il pulsante <i>Aggiungi al percorso_G</i>; • Verificare che sia stato creato il nuovo <i>percorso_G</i> personalizzato avente come primo <i>frame_G</i> il nodo scelto. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD4.4	<p>L'utente autenticato intende modificare un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Selezionare il <i>percorso_G</i> a cui intende apportare delle modifiche; • Selezionare una delle possibili modifiche da apportare; • Verificare che il <i>percorso_G</i> personalizzato selezionato sia stato modificato correttamente. 	Non Implementato
TVFD4.4.1	<p>L'utente autenticato intende aggiungere un <i>frame_G</i> ad un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Selezionare il <i>percorso_G</i> a cui intende aggiungere il <i>frame_G</i>; • Selezionare il nodo della <i>mappa mentale_G</i> che desidera aggiungere; • Premere il pulsante <i>Aggiungi al percorso_G</i>; • Verificare che sia stato aggiunto il nodo desiderato al <i>percorso_G</i> personalizzato selezionato. 	Non Implementato

Id Test	Descrizione	Stato
TVFD4.4.3	<p>L'utente autenticato intende eliminare un <i>frame_G</i> in un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Selezionare il <i>percorso_G</i> da cui intende eliminare il <i>frame_G</i>; • Premere il pulsante di eliminazione relativo al <i>frame_G</i> da eliminare; • Verificare sia stato eliminato il <i>frame_G</i> desiderato dal <i>percorso_G</i> personalizzato selezionato. 	<i>Non Implementato</i>
TVFD4.4.5	<p>L'utente autenticato intende modificare il nome di un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Selezionare il <i>percorso_G</i> a cui intende modificare il nome; • Inserire nell'apposito spazio il nuovo nome per il <i>percorso_G</i>; • Premere il pulsante <i>Salva</i> inerente al nuovo campo inserito; • Verificare che il nome del <i>percorso_G</i> selezionato sia stato modificato coerentemente con quanto specificato. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD4.5	<p>L'utente autenticato intende eliminare un <i>percorso di presentazione_G</i> personalizzato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditPathView</i>; • Selezionare il <i>percorso_G</i> che intende eliminare; • Premere il pulsante <i>Elimina percorso_G</i>; • Verificare che il <i>percorso_G</i> selezionato sia stato eliminato. 	<i>Non Implementato</i>
TVFF4.6	<p>L'utente autenticato intende scegliere le impostazioni generali dell'aspetto grafico del <i>progetto_G</i>. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Selezionare il tab <i>Impostazioni</i>; • Apportare una modifica alle impostazioni dell'aspetto grafico del <i>progetto_G</i>; • Confermare la modifica; • Verificare che al <i>progetto_G</i> sia stata apportata la modifica scelta per le impostazioni generali dell'aspetto grafico. 	<i>Non Implementato</i>
TVFF4.6.1.1	<p>L'utente autenticato intende scegliere una famiglia di font di default per il testo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Selezionare il tab <i>Impostazioni</i>; • Scegliere il nuovo font da utilizzare; • Confermare la modifica; • Verificare che sia stata il font del testo sia stato modificato coerentemente con quanto indicato. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFF4.6.1.2	<p>L'utente autenticato intende scegliere un colore di default per il testo. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella <i>EditMapView</i>;• Selezionare il tab Impostazioni;• Scegliere il nuovo colore da utilizzare per il testo;• Confermare la modifica;• Verificare che sia stata apportata la modifica al colore per il testo.	<i>Non Implementato</i>
TVFF4.6.2	<p>L'utente autenticato intende scegliere un colore di sfondo per i <i>frame_G</i> del <i>progetto_G</i>. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella <i>EditMapView</i>;• Selezionare il tab <i>Impostazioni</i>;• Scegliere il nuovo colore di sfondo per i <i>frame_G</i>;• Confermare la modifica;• Verificare che sia stata apportata la modifica al colore di sfondo per i <i>frame_G</i> del <i>progetto_G</i>.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFO5	<p>L'utente autenticato intende modificare un <i>progetto_G</i> esistente da una <i>piattaforma_G</i> desktop. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato tramite un dispositivo desktop;• Trovarsi nella <i>Dashboard</i>• Selezionare la voce di modifica del relativo <i>progetto_G</i>;• Apportare delle modifiche al <i>progetto_G</i> scelto;• Verificare che il <i>progetto_G</i> selezionato sia stato modificato coerentemente con quanto specificato.	<i>Non Implementato</i>
TVFO7	<p>L'utente autenticato intende eseguire una presentazione. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella <i>Dashboard</i>;• Premere il pulsante per visualizzare i <i>percorsi di presentazione_G</i> disponibili;• Selezionare uno tra i <i>percorsi di presentazione_G</i> proposti;• Verificare che venga avviato il <i>percorso di presentazione_G</i> selezionato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFO7.2	<p>L'utente autenticato intende spostarsi al $frame_G$ successivo nella sequenza di presentazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Aver selezionato un <i>percorso di presentazione_G</i> per il <i>progetto_G</i> desiderato; • Trovarsi nella <i>PresentationView</i>; • Premere il pulsante <i>Successiva</i> nel menu in basso; • Verificare che venga correttamente presentato il $frame_G$ successivo del <i>percorso di presentazione_G</i> selezionato. 	<i>Non Implementato</i>
TVFO7.3	<p>L'utente autenticato intende spostarsi al $frame_G$ precedente nella sequenza di presentazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Aver selezionato un <i>percorso di presentazione_G</i> per il <i>progetto_G</i> desiderato; • Trovarsi nella <i>PresentationView</i>; • Premere il pulsante <i>Precedente</i> nel menu in basso; • Verificare che venga correttamente presentato il $frame_G$ precedente del <i>percorso di presentazione_G</i> selezionato. 	<i>Non Implementato</i>
TVFD7.5	<p>L'utente autenticato intende chiudere una presentazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Aver selezionato un <i>percorso di presentazione_G</i> per il <i>progetto_G</i> desiderato; • Trovarsi nella <i>PresentationView</i>; • Premere il pulsante <i>EditPathView</i> nella barra di navigazione; • Verificare che la presentazione si sia chiusa. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD7.6	<p>L'utente autenticato intende, durante una presentazione, passare ad un $frame_G$ correlato al $frame_G$ visualizzato tramite associazione gerarchica o logica. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Aver selezionato un <i>percorso di presentazione_G</i> per il <i>progetto_G</i> desiderato; • Trovarsi nella <i>PresentationView</i>; • Selezionare all'interno di una delle due liste di nodi il nodo desiderato; • Verificare che venga correttamente presentato il $frame_G$ corrispondente al nodo selezionato. 	Non Implementato
TVFO7.7	<p>L'utente autenticato intende visionare il contenuto del $frame_G$ presente in un nodo. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Selezionare il nodo di cui si desidera visualizzare il $frame_G$; • Verificare che nell'apposita $view_G$ venga visualizzato il contenuto del $frame_G$ relativo al nodo selezionato. 	Non Implementato
TVFO8	<p>L'utente autenticato intende eseguire una presentazione da desktop. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato tramite un dispositivo desktop; • Trovarsi nella <i>Dashboard</i>; • Premere il pulsante per visualizzare i <i>percorsi di presentazione_G</i> disponibili; • Selezionare uno tra i <i>percorsi di presentazione_G</i> proposti; • Verificare che venga avviato il <i>percorso di presentazione_G</i> selezionato. 	Non Implementato

Id Test	Descrizione	Stato
TVFO9	<p>L'utente autenticato intende eseguire una presentazione da mobile. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato tramite un dispositivo mobile; • Trovarsi nella <i>Dashboard</i>; • Premere il pulsante per visualizzare i <i>percorsi di presentazione_G</i> disponibili; • Selezionare uno tra i <i>percorsi di presentazione_G</i> proposti; • Verificare che venga avviato il <i>percorso di presentazione_G</i> selezionato. 	<i>Non Implementato</i>
TVFO10	<p>L'utente autenticato vuole eliminare un <i>progetto_G</i> precedentemente creato. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>Dashboard</i>; • Premere il pulsante <i>Elimina</i> relativo al <i>progetto_G</i> che desidera eliminare; • Verificare il <i>progetto_G</i> selezionato sia stato eliminato. 	<i>Non Implementato</i>
TVFO11	<p>L'utente autenticato desidera aprire un <i>progetto_G</i>. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>Dashboard</i>; • Selezionare una delle voci tra <i>Modifica</i> e <i>Presenta</i> relative al <i>progetto_G</i> desiderato; • Verificare che il <i>progetto_G</i> desiderato sia stato aperto nella modalità indicata. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD14	<p>L'utente autenticato intende esportare in formato pdf_G la presentazione che sta visualizzando. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>PresentationView</i>; • Premere il pulsante <i>Esporta-Stampa</i> presente nella barra di navigazione; • Scegliere un nome ed un $percorso_G$ nel proprio filesystem in cui salvare il file; • Scegliere <i>Esportazione in pdf_G</i> all'interno della $view_G$ di esportazione-stampa; • Premere il pulsante <i>Salva</i> all'interno della $view_G$ di esportazione-stampa; • Verificare che il pdf_G esportato contenga la presentazione visualizzata. 	<i>Non Implementato</i>
TVFO16	<p>L'utente desidera stampare una presentazione che sta visualizzando. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>PresentationView</i>; • Premere il pulsante <i>Esporta-Stampa</i> presente nella barra di navigazione; • Scegliere la stampante all'interno della $view_G$ di esportazione-stampa; • Premere il pulsante <i>Stampa</i> all'interno della $view_G$ di esportazione-stampa; • Verificare che le stampe prodotte contengano la presentazione visualizzata. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFD17	<p>L'utente autenticato intende eseguire una presentazione non lineare. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Aver selezionato un <i>percorso di presentazione_G</i> per il <i>progetto_G</i> desiderato; • Trovarsi nella <i>PresentationView</i>; • Scegliere tramite i menù laterali il successivo <i>frame_G</i> da presentare; • Verificare che la presentazione segua le scelte effettuate. 	<i>Non Implementato</i>
TVFO18	<p>L'utente autenticato intende visualizzare un effetto grafico durante le transizioni tra <i>frame_G</i> nella presentazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Aver selezionato un <i>percorso di presentazione_G</i> per il <i>progetto_G</i> desiderato; • Trovarsi nella <i>PresentationView</i>; • Selezionare <i>Successivo</i> o <i>Precedente</i> dal menù in basso; • Verificare che durante le transizioni venga visualizzato un effetto grafico. 	<i>Non Implementato</i>
TVFD22	<p>L'utente autenticato intende creare una mappa mentale. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i>; • Partendo dal nodo radice utilizzare le varie opzioni offerte per creare la mappa; • Verificare che la <i>mappa mentale_G</i> sia stata creata in maniera corretta. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFF25	<p>L'utente autenticato intende chiudere il <i>progetto_G</i> corrente. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i> o nella <i>EditPath-View</i>; • Selezionare la voce <i>Dashboard</i> presente nel menù di navigazione; • Verificare che il <i>progetto_G</i> sia stato chiuso e che venga visualizzata la <i>Dashboard</i>. 	<i>Non Implementato</i>
TVFD26	<p>L'utente autenticato intende consultare il manuale direttamente dall'applicazione. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Essere autenticato; • Trovarsi nella <i>EditMapView</i> o nella <i>EditPath-View</i>; • Selezionare la voce <i>Manuale</i> presente nel menù di navigazione; • Verificare che il manuale venga visualizzato correttamente. 	<i>Non Implementato</i>
TVFO30	<p>L'utente intende scegliere tra registrazione e login. All'utente è richiesto di:</p> <ul style="list-style-type: none"> • Trovarsi nella <i>LoginView</i> o nella <i>Registration-View</i>; • Selezionare il link <i>Registrati</i> se si trova nella <i>LoginView</i> o il link <i>Login</i> se si trova nella <i>RegistrationView</i>; • Verificare che il link selezionato abbia caricato la <i>view_G</i> corretta. 	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TVFO32	<p>L'utente autenticato intende effettuare il logout dal sistema. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Premere il pulsante <i>Logout</i> nel menù in alto;• Verificare che l'operazione di logout vada a buon fine.	<i>Non Implementato</i>
TVFD33	<p>L'utente autenticato intende ricevere un messaggio di errore se prova a creare un nuovo <i>progetto_G</i> con nome già utilizzato per un <i>progetto_G</i> esistente. All'utente è richiesto di:</p> <ul style="list-style-type: none">• Essere autenticato;• Trovarsi nella <i>Dashboard</i>;• Premere il pulsante <i>Crea nuovo progetto_G</i>;• Inserire il nome di un <i>progetto_G</i> già esistente per il nuovo <i>progetto_G</i>;• Premere il pulsante di conferma creazione nuovo <i>progetto_G</i>;• Verificare che il sistema restituisca un messaggio di errore che motivi il fallimento dell'operazione.	<i>Non Implementato</i>

Tabella 2: Test di Validazione

4.3 Test di Sistema

Con questa tipologia di test si vuole verificare il corretto funzionamento delle componenti architetturali.

I test di sistema saranno descritti nel modo seguente:

TS[*TipoRequisito*][*ImportanzaRequisito*][*IdRequisito*]

dove:

- **TipoRequisito** può assumere valori tra:
 - *F* per i requisiti funzionali;
 - *Q* per i requisiti di qualità;
 - *V* per i requisiti di vincolo;
 - *P* per i requisiti prestazionali.
- **ImportanzaRequisito** può assumere valori tra:
 - *D* per i requisiti desiderabili;
 - *O* per i requisiti di obbligatori;
 - *F* per i requisiti di facoltativi.
- **IdRequisito** può assumere un valore gerarchico che identifica il singolo requisito.

Id Test	Descrizione	Stato
TSFO1	Viene verificato che il sistema crei correttamente un nuovo progetto.	<i>Non Implementato</i>
TSFO2	Viene verificato che il sistema crei correttamente un progetto da dispositivo desktop.	<i>Non Implementato</i>
TSFO4	Viene verificato che il sistema modifichi correttamente un progetto.	<i>Non Implementato</i>
TSFD4.1	Viene verificato che il sistema modifichi correttamente il nome di un progetto.	<i>Non Implementato</i>
TSFO4.2	Viene verificato che il sistema modifichi correttamente la struttura della mappa mentale.	<i>Non Implementato</i>
TSFO4.2.1	Viene verificato che il sistema selezioni correttamente un nodo della mappa.	<i>Non Implementato</i>
TSFO4.2.2	Viene verificato che il sistema aggiunga correttamente un nodo alla mappa.	<i>Non Implementato</i>
TSFO4.2.3	Viene verificato che il sistema modifichi correttamente il nodo desiderato della mappa.	<i>Non Implementato</i>
TSFO4.2.3.2	Viene verificato che il sistema selezioni correttamente un'immagine in un nodo della mappa.	<i>Non Implementato</i>
TSFO4.2.3.4	Viene verificato che il sistema aggiunga correttamente un elemento testuale in un nodo della mappa.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TSFO4.2.3.5	Viene verificato che il sistema aggiunga correttamente un'immagine ad un nodo della mappa.	<i>Non Implementato</i>
TSFO4.2.3.7	Viene verificato che il sistema modifichi correttamente l'elemento testuale selezionato.	<i>Non Implementato</i>
TSFO4.2.3.12	Viene verificato che il sistema elimini correttamente l'elemento testuale selezionato.	<i>Non Implementato</i>
TSFO4.2.3.13	Viene verificato che il sistema elimini correttamente l'immagine selezionata.	<i>Non Implementato</i>
TSFD4.2.3.15	Viene verificato che il sistema dia correttamente il nome desiderato al nodo selezionato.	<i>Non Implementato</i>
TSFD4.2.3.16	Viene verificato che il sistema modifichi correttamente il titolo del nodo selezionato.	<i>Non Implementato</i>
TSFD4.2.3.17	Viene verificato che il sistema selezioni correttamente il titolo di un nodo.	<i>Non Implementato</i>
TSFD4.2.3.18	Viene verificato che il sistema ridimensioni correttamente il titolo di un nodo.	<i>Non Implementato</i>
TSFD4.2.3.19	Viene verificato che il sistema ridimensioni correttamente il contenuto testuale di un nodo.	<i>Non Implementato</i>
TSFD4.2.3.20	Viene verificato che il sistema ridimensioni correttamente l'immagine presente in un nodo.	<i>Non Implementato</i>
TSFO4.2.5	Viene verificato che il sistema elimini correttamente il nodo selezionato dalla mappa.	<i>Non Implementato</i>
TSFO4.2.6	Viene verificato che il sistema crei correttamente l'associazione tra due nodi selezionati.	<i>Non Implementato</i>
TSFO4.2.7	Viene verificato che il sistema elimini correttamente l'associazione tra due nodi selezionata.	<i>Non Implementato</i>
TSFO7	Viene verificato che il sistema esegua correttamente la presentazione di un progetto.	<i>Non Implementato</i>
TSFO7.2	Viene verificato che il sistema esegua correttamente il passaggio al frame successivo in modalità presentazione.	<i>Non Implementato</i>
TSFD7.5	Viene verificato che il sistema chiuda correttamente la presentazione corrente.	<i>Non Implementato</i>
TSVO4	Viene verificato che l'applicativo funzioni su <i>Mozilla Firefox</i> versione 33.0 o superiore.	<i>Non Implementato</i>
TSVO5	Viene verificato che l'applicativo funzioni su <i>Google Chrome</i> versione 31.0 o superiore.	<i>Non Implementato</i>
TSVO10	Viene verificato che l'applicativo funzioni su <i>Google Chrome per Android</i> versione 39.0 o superiore.	<i>Non Implementato</i>

Tabella 3: Test di Sistema

4.4 Test di Integrazione

Con questa tipologia di test si vuole determinare il corretto funzionamento delle componenti progettate durante la definizione dell'architettura ad alto livello.

I test di integrazione saranno descritti nel modo seguente:

$$TI[IdComponente]$$

dove:

- **IdComponente** rappresenta il codice identificativo crescente del componente considerato.

È stato scelto di utilizzare un approccio top-down nel determinare i test di integrazione. Di seguito viene riportato un diagramma informale per rendere chiara la struttura dei test identificati.

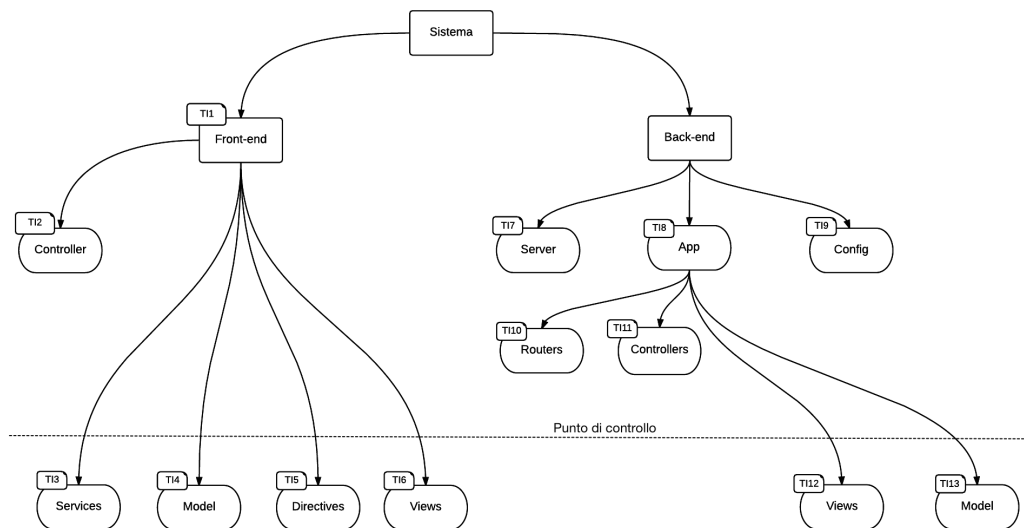


Figura 1: Albero di integrazione delle componenti

Nell'approccio top-down dei test di integrazione i moduli di livello più alto vengono sottoposti a test e integrati per primi. Così facendo anche la logica di alto livello e il flusso di dati vengono sottoposti a test fin da subito; sarà perciò necessario simulare le componenti di livello più basso con degli *stub_G*. Una volta codificate, le componenti di più basso livello dovranno a loro volta essere integrate e testate. L'approccio top-down rientra tra le strategie di integrazione incrementali, che conferiscono il vantaggio di poter determinare in modo più immediato quale componente causa problemi: i difetti rilevati dai test, infatti, nella maggioranza dei casi saranno da attribuirsi all'ultima componente aggiunta.

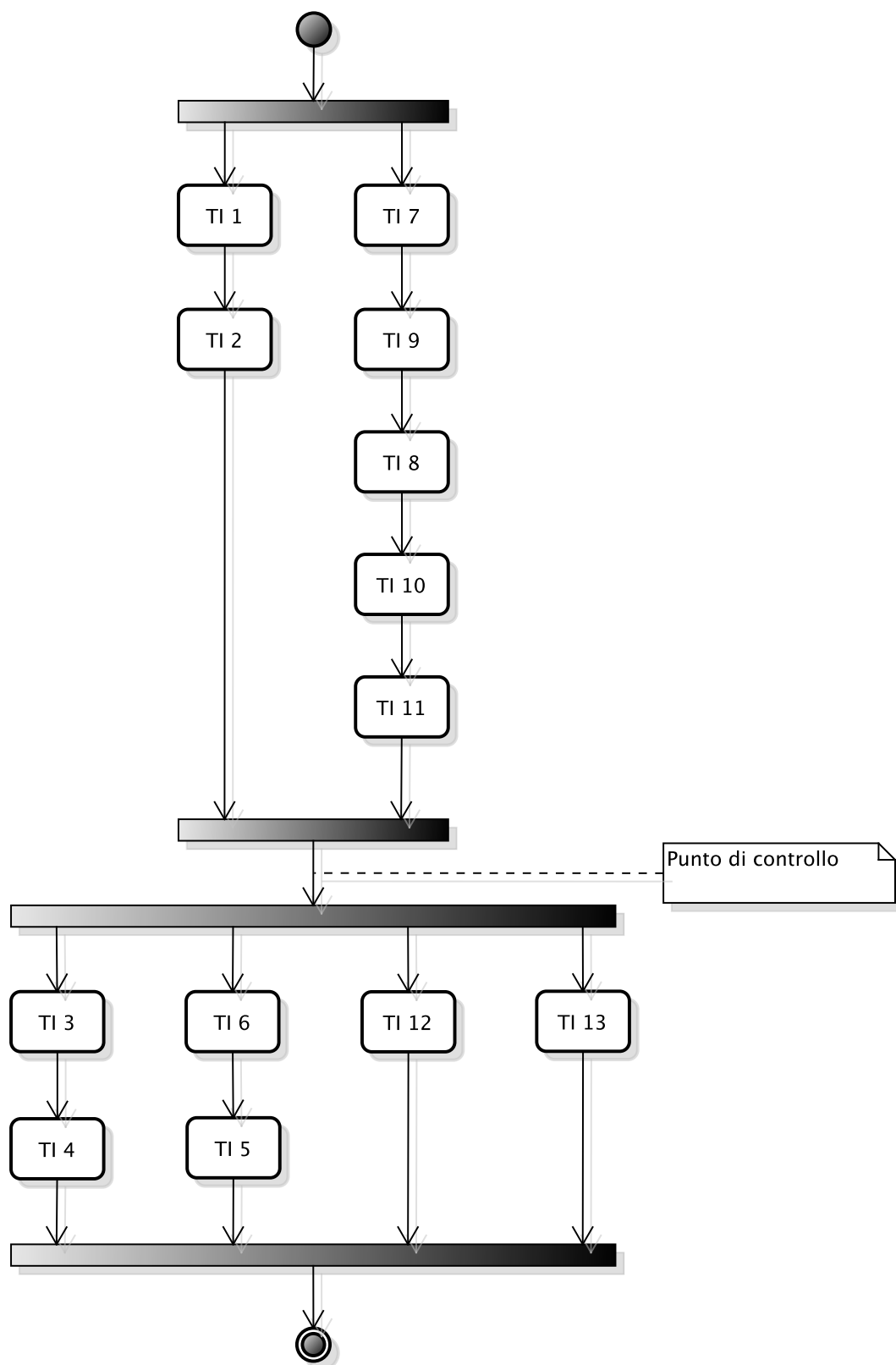


Figura 2: Diagramma di attività dei test di integrazione

Id Test	Descrizione	Stato
TI1	Viene verificato che l'applicazione Web gestisca correttamente il Front-End del prodotto e le sue interazioni con il Back-End.	<i>Non Implementato</i>
TI2	Viene verificato che i <i>Controller</i> del Front-End si integrino correttamente nell'applicazione Web.	<i>Non Implementato</i>
TI3	Viene verificato che i <i>Services</i> permettano di interagire correttamente con il Back-End.	<i>Non Implementato</i>
TI4	Viene verificato che il <i>Model</i> si integri correttamente con i <i>Services</i> e con le componenti dell'applicazione che lo utilizzano.	<i>Non Implementato</i>
TI5	Viene verificato che le <i>Directives</i> si integrino correttamente con le <i>View</i> .	<i>Non Implementato</i>
TI6	Viene verificato che le <i>View</i> si integrino correttamente con i <i>Controller</i> e che visualizzino in modo corretto i dati da essi ricevuti.	<i>Non Implementato</i>
TI7	Viene verificato che il server si avvii correttamente, utilizzando <i>Config</i> per effettuare le configurazioni dell'applicazione, e che l'applicazione Web gestisca correttamente il Back-End del prodotto in modo tale da fornire al Front-End tutte le informazioni richieste.	<i>Non Implementato</i>
TI8	Viene verificato che <i>App</i> si integri correttamente con le librerie di Node.js utilizzate.	<i>Non Implementato</i>
TI9	Viene verificato che <i>Config</i> si integri con <i>Server</i> , carichi correttamente tutte le librerie per Node.js che utilizzerà e che istanzi le classi del package <i>App</i> in modo corretto.	<i>Non Implementato</i>
TI10	Viene verificato che i <i>Routers</i> si integrino correttamente con il package <i>App</i> e reindirizzino adeguatamente le richieste REST destinate al server.	<i>Non Implementato</i>
TI11	Viene verificato che i <i>Controller</i> si integrino correttamente e gestiscano le richieste inoltrate dai <i>Routers</i> .	<i>Non Implementato</i>
TI12	Viene verificato che le <i>View</i> si integrino con i <i>Controller</i> per fornire al Front-End pagine web statiche.	<i>Non Implementato</i>
TI13	Viene verificato che il <i>Model</i> si integri correttamente con i <i>Controller</i> per la gestione dell'inserimento, della modifica e dell'eliminazione dei dati.	<i>Non Implementato</i>

Tabella 4: Test di Integrazione

4.5 Test di Unità

Con questa tipologia di test si vuole verificare il corretto funzionamento delle unità individuate durante la definizione dell'architettura ad alto livello.

I test di unità saranno descritti nel modo seguente:

TU[*IdTest*]

dove:

- **IdTest** rappresenta il codice identificativo crescente dell'unità considerata.

Id Test	Descrizione	Stato
TU1	Verificare che l'istanza del server venga creata correttamente e che si metta in ascolto su <code>localhost:3333</code> o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU2	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al <code>concreteHandler</code> corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU3	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al <code>concreteHandler</code> corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU4	Verificare che, ricevendo delle richieste conformi alle API definite, l'oggetto passi il controllo al <code>concreteHandler</code> corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU5	Verificare che, in base ai parametri forniti in input, il JSON ritornato contenga il messaggio d'errore corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU6	Verificare che, in base ai parametri forniti in input, il JSON ritornato contenga il messaggio d'errore corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU7	Verificare che, in base alle informazioni fornite in input, il metodo <i>getter</i> setti in modo corretto i parametri correlati o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU8	Verificare che il messaggio d'errore venga costruito in modo coerente rispetto ai dati passati al costruttore e che i metodi disponibili restituiscano l'informazione nel formato desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU9	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, controllando l'esistenza dell'id del nodo o dell'id dell'associazione nel database o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU10	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornata la mappa aggiungendo o eliminando associazioni fra i nodi o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU11	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornata la mappa aggiungendo, eliminando o modificando i nodi o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU12	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, controllando l'esistenza dell'id del percorso di presentazione nel database o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU13	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati i percorsi di presentazione o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU14	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti la gestione di progetti o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU15	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni riguardanti registrazione ed autenticazione dell'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU16	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, verificando che l'utente che esegue la richiesta sia effettivamente un utente autenticato e mantenendo aggiornate le informazioni riguardanti l'identificazione dell'utente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU17	Verificare che, dato in input un JSON formattato correttamente, il metodo restituisca un JSON corretto contenente le informazioni richieste riguardanti il contenuto di un nodo o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU18	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni del nodo in modo corretto, creando un nuovo nodo, aggiornandone il contenuto e restituendone le informazioni nel relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU19	Verificare che, in base ai parametri forniti in input, il metodo crei un'istanza di <code>PathModel</code> che rappresenta il percorso di presentazione di default per il progetto, restituisca il relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU20	Verificare che, in base ai parametri forniti in input, il metodo crei un'istanza di <code>PathModel</code> che rappresenta un percorso di presentazione personalizzato per il progetto, restituisca il relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU21	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornate le informazioni dell'istanza di <code>PathModel</code> in modo corretto, aggiungendo e rimuovendo nodi dal percorso di presentazione e rinominando il percorso stesso, che restituiscano il relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU22	Verificare che, in base ai parametri forniti in input, il metodo crei un nuovo progetto, completo di mappa mentale e di un percorso di presentazione di default con un nodo preesistente, restituisca il relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU23	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, mantenendo aggiornati i parametri di layout del progetto in modo corretto, restituiscano il relativo JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU24	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, aggiungendo e modificando nodi del progetto e restituendo i dati aggiornati in JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU25	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, rimuovendo nodi dal progetto e dai percorsi di presentazione in cui erano inclusi, rimuovendo le associazioni ad essi correlate e restituendo i dati aggiornati in JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU26	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, aggiungendo e rimuovendo associazioni fra i nodi e restituendo i dati aggiornati in JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU27	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, aggiungendo, rinominando, rimuovendo percorsi di presentazione e aggiungendo, rimuovendo nodi da tali percorsi e restituendo i dati aggiornati in JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU28	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, restituendo in JSON la lista dei percorsi e la lista dei nodi di un percorso specifico o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU29	Verificare che, all'inizializzazione dell'applicazione, la collection <code>relations</code> sia vuota e che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, costruendo in modo corretto <i>relazioni gerarchiche</i> ed <i>associazioni</i> e restituendo i dati aggiornati in JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU30	Verificare che, all'inizializzazione dell'applicazione, la collection <code>users</code> sia vuota e che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, costruendo un nuovo oggetto <code>Utente</code> , oppure restituendo opportuni avvisi nel caso in cui non siano stati inseriti tutti i parametri secondo indicazione o nel caso in cui la mail sia già stata utilizzata per un altro account e restituendo i dati aggiornati in JSON o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU31	Verificare che il metodo effettui le operazioni richieste, costruendo l'oggetto corrispondente e, per mezzo di <code>\$mdThemingProvider</code> , impostando i colori di default dell'applicazione o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU32	Verificare che il metodo effettui le operazioni richieste, costruendo l'oggetto corrispondente e, per mezzo di <code>\$routeProvider</code> , associando ad ogni route un controller ed una view o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU33	Verificare che il metodo effettui le operazioni richieste, costruendo l'oggetto corrispondente, il quale funge da gestore per gli eventi globali o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU34	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiAddToPath</code> , restituendo se il bottone di conferma è abilitato, gestendo correttamente gli eventi di click dei pulsanti di conferma ed annullamento dell'operazione di inserimento di un nodo nel percorso di presentazione o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU35	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiAssociationAdder</code> , restituendo se il bottone di conferma è abilitato, gestendo correttamente gli eventi di click dei pulsanti di conferma ed annullamento dell'operazione di inserimento di un'associazione o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU36	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiContextMenu</code> , invocando correttamente la funzione <code>callback</code> del parametro <code>fn</code> , nascondendo il menù per mezzo del metodo <code>\$mdDialog.cancel</code> o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU37	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, creando l'oggetto corrispondente e popolando lo <code>\$scope</code> nel modo atteso o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU38	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, aggiornando i progetti dell'utente, che <code>\$scope</code> venga modificato nel modo atteso e che, in caso di caricamento di altre view, venga effettuata correttamente la chiamata a <code>\$location</code> o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU39	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiEditableNodeContent</code> , gestendo correttamente la selezione del contenuto di un nodo sollevando l'evento <code>nodecontent-selected</code> e fornendo l'id dell'oggetto <code>NodeContent</code> o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU40	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiErrorMessage</code> , nascondendo tale directive o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU41	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiHeader</code> in modo corretto o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU42	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, controllando se l'URL corrente contiene la stringa ricevuta come parametro, ritornando il titolo da visualizzare, adattando le dimensioni della mappa mentale e permettendo di regolare lo zoom su di essa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU43	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, reindirizzando l'utente alla view identificata dall'URL ricevuto come parametro, visualizzando la directive per la modifica delle impostazioni del progetto, invocando la funzionalità di stampa del browser, gestendo gli eventi <code>click</code> sul pulsante per modificare la mappa mentale, sul pulsante per modificare i percorsi di presentazione, sul pulsante di logout o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU44	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della visibilità della directive <code>premiHierarchicalMenu</code> , rendendo visibile o nascondendo la directive (utilizzando <code>\$mdUtil</code> e <code>\$mdSidenav</code>), controllando se la directive è visibile (utilizzando <code>\$mdSidenav</code>) o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU45	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica applicativa del <i>login</i> , verificando se è già presente una sessione, recuperando i dati dallo <code>\$scope</code> ed effettuando l'autenticazione (utilizzando <code>AuthenticationService</code>), effettuando il redirect verso la pagina di registrazione o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU46	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, creando l'oggetto corrispondente e popolando lo <code>\$scope</code> nel modo atteso o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU47	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, editando la mappa secondo le indicazioni e popolando lo <code>\$scope</code> nel modo atteso o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU48	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, selezionando l'elemento desiderato, aggiornando lo <code>\$scope</code> nel modo atteso e generando correttamente gli eventi adatti o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU49	Verificare che i metodi effettuino le operazioni richieste, resettando l'elemento desiderato e aggiornando lo <code>\$scope</code> nel modo atteso o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU50	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica legata alla modifica del contenuto di un nodo, aggiungendo o rimuovendo elementi (come immagini e testo), gestendo gli eventi <code>click</code> di conferma o annullamento delle modifiche o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU51	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, costruendo l'oggetto gestore della directive <code>premiNode</code> , utilizzato per gestire lo stile di visualizzazione del frame che deve essere visualizzato dalla directive o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU52	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica e della visibilità della directive <code>premiPath</code> , rimuovendo un nodo dal percorso di presentazione, salvando le modifiche attuate o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU53	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore dei percorsi di presentazione, ritornando la stringa da utilizzare come etichetta segnaposto e ritornando un booleano rappresentante lo stato del menù o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU54	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, nascondendo o rendendo visibile il menù laterale, impostando il percorso corrente come riferimento al parametro <code>path</code> , deselegionando il percorso di presentazione precedentemente selezionato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU55	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, gestendo l'evento di aggiunta di un nodo al percorso di presentazione, l'evento per interrompere l'ascolto degli eventi di <code>MindmapService</code> , gli eventi sollevati da <code>MindmapService</code> per selezionare e deselegionare il nodo corrente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU56	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica applicativa della directive <code>premiPathsList</code> , utilizzando opportunamente <code>PathService</code> per creare un nuovo percorso di presentazione, rimuoverne uno già definito, confermare la selezione e avviare la presentazione di un percorso specifico o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU57	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica applicativa della presentazione di un percorso, visualizzando la slide di indice <code>index</code> della sequenza del percorso di presentazione, la slide contenente il nodo avente <code>id</code> uguale a <code>nodeId</code> , la slide precedente, la slide successiva, ritornando le classi CSS da associare alla slide che contiene l'oggetto <code>Node</code> ricevuto come parametro o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU58	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica applicativa della direttiva <code>premiPresentationFooter</code> , permettendo di abilitare la modalità <i>schermo intero</i> e di regolamentare tutti i meccanismi per visualizzare e nascondere il footer, sia in modo istantaneo che in seguito a timeout o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU59	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, costruendo l'oggetto gestore della logica applicativa riguardante l'esecuzione di un percorso di presentazione o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU60	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di passare da un frame ad un frame correlato, emettendo l'evento <code>presentation-previousStep</code> al passaggio al frame precedente e <code>presentation-nextStep</code> al passaggio al frame successivo, aggiornando le informazioni al cambio del nodo, visualizzando la pagina di stampa, permettendo di ripristinare la presentazione lineare o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU61	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, permettendo di gestire i meccanismi per visualizzare e nascondere i menù, sia in modo istantaneo che in seguito a timeout, visualizzando se i menù sono visualizzati o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU62	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica di funzionamento della directive <code>premiProjectSettingsEditor</code> , aggiornando l'anteprima dello stile del progetto, gestendo gli eventi generati dal click dei tasti per salvare o dimenticare le modifiche apportate o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU63	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica di funzionamento della directive <code>premiProjectsList</code> , rendendo visibile la finestra di dialogo per la scelta del percorso di presentazione da presentare o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU64	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della logica di funzionamento della registrazione di un utente, utilizzando <code>AuthenticationService</code> per registrare un nuovo utente in caso di conferma, effettuando il redirect verso la pagina di <i>login</i> o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU65	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto gestore della visibilità della directive <code>premiSmartMenu</code> , utilizzando <code>\$mdUtil</code> e <code>\$mdSidenav</code> per settare la visibilità della directive e controllare se risulta visibile o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU66	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, costruendo l'oggetto contenente le informazioni sull'errore, ritornando correttamente il titolo, il messaggio ed il codice dell'errore o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU67	Verificare che i metodi <i>getter</i> effettuino le operazioni richieste, restituendo le informazioni sul nodo desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU68	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, impostando e/o eliminando parte del contenuto del nodo desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU69	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, creando l'oggetto corrispondente, settando il contenuto del nodo nel modo atteso e restituendo le informazioni in esso contenute o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU70	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, settando le informazioni relative al nodo o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU71	Verificare che i metodi <i>getter</i> effettuino le operazioni richieste, restituendo le informazioni su un passo del percorso di presentazione desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU72	Verificare che, in base ai parametri forniti in input, i metodi effettuino le operazioni richieste, aggiungendo ed eliminando un passo di presentazione dal percorso di presentazione desiderato prestabilito o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU73	Verificare che i metodi <i>getter</i> effettuino le operazioni richieste, restituendo le informazioni sugli attributi del percorso di presentazione desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU74	Verificare che, in base ai parametri forniti in input, il metodo effettui le operazioni richieste, settando il nome del percorso di presentazione desiderato, verificando se è il percorso di presentazione di default o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU75	Verificare che i metodi <i>getter</i> effettuino le operazioni richieste, restituendo le informazioni sui nodi associati a quello desiderato o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU76	Verificare che i metodi <i>getter</i> effettuino le operazioni richieste, restituendo le informazioni sul progetto corrente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU77	Verificare che, in base ai parametri forniti in input, i metodi <i>setter</i> effettuino le operazioni richieste, settando le informazioni del progetto corrente o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Id Test	Descrizione	Stato
TU78	Verificare che, in base ai parametri forniti in input, utilizzando il service <code>\$http</code> , i metodi effettuino le richieste riguardanti l'autenticazione al backend in maniera corretta, che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>
TU79	Verificare che, in base ai parametri forniti in input, i metodi si occupino di esaudire le modifiche richieste utilizzando le funzionalità offerte da <i>Cytoscape.js</i> , che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU80	Verificare, utilizzando il service <code>\$httpBackend</code> fornito da <i>Angular.js</i> , che, in base ai parametri forniti in input, i metodi richiedano in maniera corretta l'aggiornamento delle informazioni inerenti la mappa mentale sia al backend che in locale, utilizzando i service <code>\$http</code> e <code>MindmapAdapterService</code> , che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU81	Verificare che, in base ai parametri forniti in input, utilizzando il service <code>MindmapAdapterService</code> , i metodi richiedano in maniera corretta l'aggiornamento delle informazioni inerenti la mappa mentale in locale, che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU82	Verificare, utilizzando il service <code>\$httpBackend</code> fornito da <i>Angular.js</i> , che, in base ai parametri forniti in input, i metodi richiedano in maniera corretta l'aggiornamento delle informazioni inerenti il percorso di presentazione al backend, utilizzando il service <code>\$http</code> , che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>
TU83	Verificare, utilizzando il service <code>\$httpBackend</code> fornito da <i>Angular.js</i> , che, in base ai parametri forniti in input, i metodi <i>getter</i> richiedano in maniera corretta le informazioni inerenti il percorso di presentazione al backend, utilizzando il service <code>\$http</code> , che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Non Implementato</i>

Id Test	Descrizione	Stato
TU84	Verificare, utilizzando il service <code>\$httpBackend</code> fornito da <i>Angular.js</i> , che, in base ai parametri forniti in input, i metodi richiedano in maniera corretta l'aggiornamento delle informazioni inerenti i progetti creati dall'utente, utilizzando il service <code>\$http</code> , che i dati vengano restituiti nella maniera attesa o che, in caso di errore, la risposta riporti lo stato anomalo riscontrato.	<i>Superato</i>

Tabella 5: Test di Unità

5 Tracciamenti dei test

5.1 Tracciamento Test di Validazione-Requisiti

Test	Requisito
TVFO1	RFO1
TVFO2	RFO2
TVFO4	RFO4
TVFD4.1	RFD4.1
TVFO4.2	RFO4.2
TVFO4.2.3	RFO4.2.3
TVFD4.3	RFD4.3
TVFD4.3.2	RFD4.3.2
TVFD4.4	RFD4.4
TVFD4.4.1	RFD4.4.1
TVFD4.4.3	RFD4.4.3
TVFD4.4.5	RFD4.4.5
TVFD4.5	RFD4.5
TVFF4.6	RFF4.6
TVFF4.6.1.1	RFF4.6.1.1
TVFF4.6.1.2	RFF4.6.1.2
TVFF4.6.2	RFF4.6.2
TVFO5	RFO5
TVFO7	RFO7
TVFO7.2	RFO7.2
TVFO7.3	RFO7.3
TVFD7.5	RFD7.5
TVFD7.6	RFD7.6
TVFO7.7	RFO7.7
TVFO8	RFO8
TVFO9	RFO9
TVFO10	RFO10
TVFO11	RFO11
TVFD14	RFD14
TVFO16	RFO16
TVFD17	RFD17
TVFO18	RFO18
TVFD22	RFD22
TVFF25	RFF25
TVFD26	RFD26
TVFO30	RFO30
TVFO32	RFO32
TVFD33	RFD33

Tabella 6: Tracciamento Test di Validazione-Requisiti

5.2 Tracciamento Requisiti-Test di Validazione

Requisito	Test
RFO1	TVFO1
RFO2	TVFO2
RFO4	TVFO4
RFD4.1	TVFD4.1
RFO4.2	TVFO4.2
RFO4.2.3	TVFO4.2.3
RFD4.3	TVFD4.3
RFD4.3.2	TVFD4.3.2
RFD4.4	TVFD4.4
RFD4.4.1	TVFD4.4.1
RFD4.4.3	TVFD4.4.3
RFD4.4.5	TVFD4.4.5
RFD4.5	TVFD4.5
RFF4.6	TVFF4.6
RFF4.6.1.1	TVFF4.6.1.1
RFF4.6.1.2	TVFF4.6.1.2
RFF4.6.2	TVFF4.6.2
RFO5	TVFO5
RFO7	TVFO7
RFO7.2	TVFO7.2
RFO7.3	TVFO7.3
RFD7.5	TVFD7.5
RFD7.6	TVFD7.6
RFO7.7	TVFO7.7
RFO8	TVFO8
RFO9	TVFO9
RFO10	TVFO10
RFO11	TVFO11
RFD14	TVFD14
RFO16	TVFO16
RFD17	TVFD17
RFO18	TVFO18
RFD22	TVFD22
RFF25	TVFF25
RFD26	TVFD26
RFO30	TVFO30
RFO32	TVFO32
RFD33	TVFD33

Tabella 7: Tracciamento Requisiti-Test di Validazione

5.3 Tracciamento Test di Sistema-Requisiti

Test	Requisito
TSFO1	RFO1
TSFO2	RFO2
TSFO4	RFO4
TSFD4.1	RFD4.1
TSFO4.2	RFO4.2
TSFO4.2.1	RFO4.2.1
TSFO4.2.2	RFO4.2.2
TSFO4.2.3	RFO4.2.3
TSFO4.2.3.2	RFO4.2.3.2
TSFO4.2.3.4	RFO4.2.3.4
TSFO4.2.3.5	RFO4.2.3.5
TSFO4.2.3.7	RFO4.2.3.7
TSFO4.2.3.12	RFO4.2.3.12
TSFO4.2.3.13	RFO4.2.3.13
TSFD4.2.3.15	RFD4.2.3.15
TSFD4.2.3.16	RFD4.2.3.16
TSFD4.2.3.17	RFD4.2.3.17
TSFD4.2.3.18	RFD4.2.3.18
TSFD4.2.3.19	RFD4.2.3.19
TSFD4.2.3.20	RFD4.2.3.20
TSFO4.2.5	RFO4.2.5
TSFO4.2.6	RFO4.2.6
TSFO4.2.7	RFO4.2.7
TSFO7	RFO7
TSFO7.2	RFO7.2
TSFD7.5	RFD7.5
TSVO4	RVO4
TSVO5	RVO5
TSVO10	RVO10

Tabella 8: Tracciamento Test di Sistema-Requisiti

5.4 Tracciamento Requisiti-Test di Sistema

Requisito	Test
RFO1	TVFO1
RFO2	TVFO2
RFO4	TVFO4
RFD4.1	TVFD4.1
RFO4.2	TVFO4.2
RFO4.2.1	TVFO4.2.1
RFO4.2.2	TVFO4.2.2
RFO4.2.3	TVFO4.2.3
RFO4.2.3.2	TVFO4.2.3.2
RFO4.2.3.4	TVFO4.2.3.4
RFO4.2.3.5	TVFO4.2.3.5
RFO4.2.3.7	TVFO4.2.3.7
RFO4.2.3.12	TVFO4.2.3.12
RFO4.2.3.13	TVFO4.2.3.13
RFD4.2.3.15	TVFD4.2.3.15
RFD4.2.3.16	TVFD4.2.3.16
RFD4.2.3.17	TVFD4.2.3.17
RFD4.2.3.18	TVFD4.2.3.18
RFD4.2.3.19	TVFD4.2.3.19
RFD4.2.3.20	TVFD4.2.3.20
RFO4.2.5	TVFO4.2.5
RFO4.2.6	TVFO4.2.6
RFO4.2.7	TVFO4.2.7
RFO7	TVFO7
RFO7.2	TVFO7.2
RFD7.5	TVFD7.5
RVO4	TVVO4
RVO5	TVVO5
RVO10	TVVO10

Tabella 9: Tracciamento Requisiti-Test di Sistema

5.5 Tracciamento Test di Integrazione-Componenti

Test	Componente
TI1	Premi::Front-End
TI2	Premi::Front-End::Controllers
TI3	Premi::Front-End::Services
TI4	Premi::Front-End::Model
TI5	Premi::Front-End::Directives
TI6	Premi::Front-End::Views
TI7	Premi::Back-End
TI8	Premi::Back-End::App
TI9	Premi::Back-End::Config
TI10	Premi::Back-End::App::Routers
TI11	Premi::Back-End::App::Controllers
TI12	Premi::Back-End::App::Views
TI13	Premi::Back-End::App::Models

Tabella 10: Tracciamento Test di Integrazione-Componenti

5.6 Tracciamento Componenti-Test di Integrazione

Componente	Test
Premi::Back-End	TI7
Premi::Back-End::App	TI8
Premi::Back-End::App::Controllers	TI11
Premi::Back-End::App::Models	TI13
Premi::Back-End::App::Routers	TI10
Premi::Back-End::App::Views	TI12
Premi::Back-End::Config	TI9
Premi::Front-End	TI1
Premi::Front-End::Controllers	TI2
Premi::Front-End::Directives	TI5
Premi::Front-End::Model	TI4
Premi::Front-End::Services	TI3
Premi::Front-End::Views	TI6

Tabella 11: Tracciamento Componenti-Test di Integrazione

5.7 Tracciamento Test di Unità-Metodi

Test	Metodi
TU1	Premi::Back-End::Server::start()
TU2	Premi::Back-End::App::Routers::ProjectRouter::- router()
TU3	Premi::Back-End::App::Routers::StaticRouter::- router()
TU4	Premi::Back-End::App::Routers::UserRouter::- router()
TU5	Premi::Back-End::App::Controllers::- ErrorHandler::handle()
TU6	Premi::Back-End::App::Controllers::- NotFoundHandler::handle()
TU7	Premi::Back-End::App::Controllers::- StaticController::getUserManual()
TU8	Premi::Back-End::App::Controllers::Errors::- PremiError::PremiError() Premi::Back-End::App::Controllers::Errors::- PremiError::toJSON() Premi::Back-End::App::Controllers::Errors::- PremiError::toString()
TU9	Premi::Back-End::App::Controllers::Projects::- NodeController::associationById() Premi::Back-End::App::Controllers::Projects::- NodeController::nodeById()
TU10	Premi::Back-End::App::Controllers::Projects::- NodeController::addAssociation() Premi::Back-End::App::Controllers::Projects::- NodeController::deleteAssociation()
TU11	Premi::Back-End::App::Controllers::Projects::- NodeController::addNode() Premi::Back-End::App::Controllers::Projects::- NodeController::deleteNode() Premi::Back-End::App::Controllers::Projects::- NodeController::updateNode()
TU12	Premi::Back-End::App::Controllers::Projects::- PathController::pathById()
TU13	Premi::Back-End::App::Controllers::Projects::- PathController::addNodeToPath() Premi::Back-End::App::Controllers::Projects::- PathController::addPath() Premi::Back-End::App::Controllers::Projects::- PathController::deleteNodeFromPath() Premi::Back-End::App::Controllers::Projects::- PathController::deletePath()

Test	Metodi
	Premi::Back-End::App::Controllers::Projects::- PathController::getPath() Premi::Back-End::App::Controllers::Projects::- PathController::updatePath()
TU14	Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::addProject() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::deleteProject() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::getAllPaths() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::getAllProjects() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::getPresentation() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::getProject() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::projectById() Premi::Back-End::App::Controllers::Projects::- ProjectManagementController::updateProject()
TU15	Premi::Back-End::App::Controllers::Users::- AuthenticationController::signin() Premi::Back-End::App::Controllers::Users::- AuthenticationController::signout() Premi::Back-End::App::Controllers::Users::- AuthenticationController::signup()
TU16	Premi::Back-End::App::Controllers::Users::- AuthorizationController::requiresLogin()
TU17	Premi::Back-End::App::Models::NodeContentModel::- <u>createNodeContent</u> ()
TU18	Premi::Back-End::App::Models::NodeModel::- getContents() Premi::Back-End::App::Models::NodeModel::- updateContents() Premi::Back-End::App::Models::NodeModel::- <u>createNode</u> ()
TU19	Premi::Back-End::App::Models::PathModel::- <u>createPath</u> ()
TU20	Premi::Back-End::App::Models::PathModel::- <u>createPath</u> ()
TU21	Premi::Back-End::App::Models::PathModel::- addNode() Premi::Back-End::App::Models::PathModel::- removeNode() Premi::Back-End::App::Models::PathModel::- setName()

Test	Metodi
TU22	Premi::Back-End::App::Models::ProjectModel::- <u>createProject()</u>
TU23	Premi::Back-End::App::Models::ProjectModel::- setBkgColor() Premi::Back-End::App::Models::ProjectModel::- setFontColor() Premi::Back-End::App::Models::ProjectModel::- setFontFamily() Premi::Back-End::App::Models::ProjectModel::- setName()
TU24	Premi::Back-End::App::Models::ProjectModel::- addNode() Premi::Back-End::App::Models::ProjectModel::- updateNode()
TU25	Premi::Back-End::App::Models::ProjectModel::- removeNode()
TU26	Premi::Back-End::App::Models::ProjectModel::- addAssociation() Premi::Back-End::App::Models::ProjectModel::- removeAssociation()
TU27	Premi::Back-End::App::Models::ProjectModel::- addNodeToPath() Premi::Back-End::App::Models::ProjectModel::- addPath() Premi::Back-End::App::Models::ProjectModel::- removeNodeFromPath() Premi::Back-End::App::Models::ProjectModel::- removePath() Premi::Back-End::App::Models::ProjectModel::- setPathName()
TU28	Premi::Back-End::App::Models::ProjectModel::- getPathList() Premi::Back-End::App::Models::ProjectModel::- getPathNodes()
TU29	Premi::Back-End::App::Models::RelationModel::- <u>createRelation()</u>
TU30	Premi::Back-End::App::Models::UserModel::- <u>createUser()</u>
TU31	Premi::Front-End::AppConfig::- AppConfig()
TU32	Premi::Front-End::AppRouter::- AppRouter()
TU33	Premi::Front-End::AppRun::AppRun()
TU34	Premi::Front-End::Controllers::- AddToPathController::addButtonDisabled()

Test	Metodi
	<pre>Premi::Front-End::Controllers::- AddToPathController::addClicked() Premi::Front-End::Controllers::- AddToPathController::AddToPathController() Premi::Front-End::Controllers::- AddToPathController::cancelClicked()</pre>
TU35	<pre>Premi::Front-End::Controllers::- AssociationAdderController::- AssociationAdderController() Premi::Front-End::Controllers::- AssociationAdderController::cancelClicked() Premi::Front-End::Controllers::- AssociationAdderController::confirmClicked() Premi::Front-End::Controllers::- AssociationAdderController::confirmDisabled()</pre>
TU36	<pre>Premi::Front-End::Controllers::- ContextMenuController::call() Premi::Front-End::Controllers::- ContextMenuController::cancel() Premi::Front-End::Controllers::- ContextMenuController::ContextMenuController()</pre>
TU37	<pre>Premi::Front-End::Controllers::- DashboardController::DashboardController()</pre>
TU38	<pre>Premi::Front-End::Controllers::- DashboardController::deleteProjectClicked() Premi::Front-End::Controllers::- DashboardController::editProjectClicked() Premi::Front-End::Controllers::- DashboardController::newProjectClicked() Premi::Front-End::Controllers::- DashboardController::startPresentationClicked()</pre>
TU39	<pre>Premi::Front-End::Controllers::- EditableNodeContentController::- EditableNodeContentController() Premi::Front-End::Controllers::- EditableNodeContentController::selected()</pre>
TU40	<pre>Premi::Front-End::Controllers::- ErrorMessageController::ErrorMessageController() Premi::Front-End::Controllers::- ErrorMessageController::hide()</pre>
TU41	<pre>Premi::Front-End::Controllers::HeaderController::- HeaderController()</pre>
TU42	<pre>Premi::Front-End::Controllers::HeaderController::- currentLocationContains() Premi::Front-End::Controllers::HeaderController::- fitMindmap()</pre>

Test	Metodi
	<pre>Premi::Front-End::Controllers::HeaderController::- getTitle() Premi::Front-End::Controllers::HeaderController::- zoomMindmap()</pre>
TU43	<pre>Premi::Front-End::Controllers::HeaderController::- logout() Premi::Front-End::Controllers::HeaderController::- mapClicked() Premi::Front-End::Controllers::HeaderController::- pathClicked() Premi::Front-End::Controllers::HeaderController::- printPage() Premi::Front-End::Controllers::HeaderController::- setCurrentLocation() Premi::Front-End::Controllers::HeaderController::- showProjectSettingsEditor()</pre>
TU44	<pre>Premi::Front-End::Controllers::- HierarchicalMenuController::close() Premi::Front-End::Controllers::- HierarchicalMenuController::- HierarchicalMenuController() Premi::Front-End::Controllers::- HierarchicalMenuController::isOpen() Premi::Front-End::Controllers::- HierarchicalMenuController::open()</pre>
TU45	<pre>Premi::Front-End::Controllers::LoginController::- login() Premi::Front-End::Controllers::LoginController::- LoginController() Premi::Front-End::Controllers::LoginController::- signUp()</pre>
TU46	<pre>Premi::Front-End::Controllers::- MindmapEditorController::MindmapEditorController()</pre>
TU47	<pre>Premi::Front-End::Controllers::- MindmapEditorController::addAssociationClicked() Premi::Front-End::Controllers::- MindmapEditorController::addNodeClicked() Premi::Front-End::Controllers::- MindmapEditorController::deleteNodeClicked() Premi::Front-End::Controllers::- MindmapEditorController::deleteRelationClicked()</pre>
TU48	<pre>Premi::Front-End::Controllers::- MindmapEditorController::nodeSelected() Premi::Front-End::Controllers::- MindmapEditorController::relationSelected()</pre>

Test	Metodi
TU49	Premi::Front-End::Controllers:- MindmapEditorController::resetNodeEdit()
TU50	Premi::Front-End::Controllers:- NodeContentsEditorController::addImageContent() Preni::Front-End::Controllers:- NodeContentsEditorController::addTextContent() Preni::Front-End::Controllers:- NodeContentsEditorController::annullaClicked() Preni::Front-End::Controllers:- NodeContentsEditorController::deleteNodeContent() Preni::Front-End::Controllers:- NodeContentsEditorController:- NodeContentsEditorController() Preni::Front-End::Controllers:- NodeContentsEditorController::salvaClicked()
TU51	Premi::Front-End::Controllers::NodeController:- NodeController()
TU52	Premi::Front-End::Controllers::PathController:- PathController() Preni::Front-End::Controllers::PathController:- removeNode() Preni::Front-End::Controllers::PathController:- saveChanges()
TU53	Preni::Front-End::Controllers:- PathsEditorController::getTongueName() Preni::Front-End::Controllers:- PathsEditorController::isOpen() Preni::Front-End::Controllers:- PathsEditorController::PathsEditorController()
TU54	Preni::Front-End::Controllers:- PathsEditorController::close() Preni::Front-End::Controllers:- PathsEditorController::deselectPath() Preni::Front-End::Controllers:- PathsEditorController::open() Preni::Front-End::Controllers:- PathsEditorController::selectPath()
TU55	Preni::Front-End::Controllers:- PathsEditorController::addNodeToPathClicked() Preni::Front-End::Controllers:- PathsEditorController::destructor() Preni::Front-End::Controllers:- PathsEditorController::nodeDeselected() Preni::Front-End::Controllers:- PathsEditorController::nodeSelected()

Test	Metodi
TU56	<pre> Premi::Front-End::Controllers::- PathsListController::createNewPath() Premi::Front-End::Controllers::- PathsListController::deletePath() Premi::Front-End::Controllers::- PathsListController::pathSelected() Premi::Front-End::Controllers::- PathsListController::PathsListController() Premi::Front-End::Controllers::- PathsListController::present() </pre>
TU57	<pre> Premi::Front-End::Controllers::- PresentationController::classForNode() Premi::Front-End::Controllers::- PresentationController::nextStep() Premi::Front-End::Controllers::- PresentationController::PresentationController() Premi::Front-End::Controllers::- PresentationController::previousStep() Premi::Front-End::Controllers::- PresentationController::setCurrentStepFromId() Premi::Front-End::Controllers::- PresentationController::setCurrentStepFromIndex() </pre>
TU58	<pre> Premi::Front-End::Controllers::- PresentationFooterController::leaveEvent() Premi::Front-End::Controllers::- PresentationFooterController::- PresentationFooterController() Premi::Front-End::Controllers::- PresentationFooterController::setFullscreen() Premi::Front-End::Controllers::- PresentationFooterController::showFooter() Premi::Front-End::Controllers::- PresentationFooterController::- stopWaitingForFooterHiding() </pre>
TU59	<pre> Premi::Front-End::Controllers::- PresentationViewerController::- PresentationViewerController() </pre>
TU60	<pre> Premi::Front-End::Controllers::- PresentationViewerController::jumpToNode() Premi::Front-End::Controllers::- PresentationViewerController::nextStep() Premi::Front-End::Controllers::- PresentationViewerController::nodeChanged() Premi::Front-End::Controllers::- PresentationViewerController::previousStep() </pre>

Test	Metodi
	<pre>Premi::Front-End::Controllers::- PresentationViewController::printPresentation() Premi::Front-End::Controllers::- PresentationViewController::resumePresentation()</pre>
TU61	<pre>Premi::Front-End::Controllers::- PresentationViewController::closeMenus() Premi::Front-End::Controllers::- PresentationViewController::isMenuOpen() Premi::Front-End::Controllers::- PresentationViewController::leaveMenu() Premi::Front-End::Controllers::- PresentationViewController::showMenus() Premi::Front-End::Controllers::- PresentationViewController::- stopWaitingForMenusHiding()</pre>
TU62	<pre>Premi::Front-End::Controllers::- ProjectSettingsEditorController::cancel() Premi::Front-End::Controllers::- ProjectSettingsEditorController::confirm() Premi::Front-End::Controllers::- ProjectSettingsEditorController::- ProjectSettingsEditorController() Premi::Front-End::Controllers::- ProjectSettingsEditorController::selectChanged()</pre>
TU63	<pre>Premi::Front-End::Controllers::- ProjectsListController::ProjectsListController() Premi::Front-End::Controllers::- ProjectsListController::showPaths()</pre>
TU64	<pre>Premi::Front-End::Controllers::- RegistrationController::createAccount() Premi::Front-End::Controllers::- RegistrationController::login() Premi::Front-End::Controllers::- RegistrationController::RegistrationController()</pre>
TU65	<pre>Premi::Front-End::Controllers::- SmartMenuController::close() Premi::Front-End::Controllers::- SmartMenuController::isOpen() Premi::Front-End::Controllers::- SmartMenuController::open() Premi::Front-End::Controllers::- SmartMenuController::SmartMenuController()</pre>
TU66	<pre>Premi::Front-End::Model::ErrorInfo::- ErrorInfo() Premi::Front-End::Model::ErrorInfo::- getCode()</pre>

Test	Metodi
	<pre>Premi::Front-End::Model::ErrorInfo::- getMessage() Premi::Front-End::Model::ErrorInfo::- getTitle()</pre>
TU67	<pre>Premi::Front-End::Model::Node::- getContents() Premi::Front-End::Model::Node::- getTitle()</pre>
TU68	<pre>Premi::Front-End::Model::Node::- addImage() Premi::Front-End::Model::Node::- addText() Premi::Front-End::Model::Node::- getContent() Premi::Front-End::Model::Node::- removeContent()</pre>
TU69	<pre>Premi::Front-End::Model::NodeContent::- getContent() Premi::Front-End::Model::NodeContent::- getHeight() Premi::Front-End::Model::NodeContent::- getId() Premi::Front-End::Model::NodeContent::- getType() Premi::Front-End::Model::NodeContent::- getWidth() Premi::Front-End::Model::NodeContent::- getX() Premi::Front-End::Model::NodeContent::- getY() Premi::Front-End::Model::NodeContent::- NodeContent()</pre>
TU70	<pre>Premi::Front-End::Model::NodeContent::- getStyle() Premi::Front-End::Model::NodeContent::- setHeight() Premi::Front-End::Model::NodeContent::- setWidth() Premi::Front-End::Model::NodeContent::- setX() Premi::Front-End::Model::NodeContent::- setY()</pre>
TU71	<pre>Premi::Front-End::Model::NodeReference::- getId() Premi::Front-End::Model::NodeReference::- getTitle()</pre>

Test	Metodi
TU72	<pre>Premi::Front-End::Model::Path::- addStep() Premi::Front-End::Model::Path::- deleteStep()</pre>
TU73	<pre>Premi::Front-End::Model::Path::getId() Premi::Front-End::Model::Path::- getName() Premi::Front-End::Model::Path::- getSteps()</pre>
TU74	<pre>Premi::Front-End::Model::Path::- isDefault() Premi::Front-End::Model::Path::- setName()</pre>
TU75	<pre>Premi::Front-End::Model::PresentationNode::- getAssociatedNodes() Premi::Front-End::Model::PresentationNode::- getChildNodes() Premi::Front-End::Model::PresentationNode::- getParentNode()</pre>
TU76	<pre>Premi::Front-End::Model::Project::- getBackgroundColor() Premi::Front-End::Model::Project::- getFontFamily() Premi::Front-End::Model::Project::- getId() Premi::Front-End::Model::Project::- getName() Premi::Front-End::Model::Project::- getRootId() Premi::Front-End::Model::Project::- getTextColor()</pre>
TU77	<pre>Premi::Front-End::Model::Project::- setBackgroundColor() Premi::Front-End::Model::Project::- setFontFamily() Premi::Front-End::Model::Project::- setName() Premi::Front-End::Model::Project::- setTextColor()</pre>
TU78	<pre>Premi::Front-End::Services::AuthenticationService::- login() Premi::Front-End::Services::AuthenticationService::- logout() Premi::Front-End::Services::AuthenticationService::- signUp()</pre>

Test	Metodi
TU79	<pre> Premi::Front-End::Services::MindmapAdapterService::- addAssociation() Premi::Front-End::Services::MindmapAdapterService::- addNode() Premi::Front-End::Services::MindmapAdapterService::- deleteAssociation() Premi::Front-End::Services::MindmapAdapterService::- drawMap() Premi::Front-End::Services::MindmapAdapterService::- getNode() Premi::Front-End::Services::MindmapAdapterService::- listen() Premi::Front-End::Services::MindmapAdapterService::- loadMap() Premi::Front-End::Services::MindmapAdapterService::- setNode() Premi::Front-End::Services::MindmapAdapterService::- stopListen() </pre>
TU80	<pre> Premi::Front-End::Services::MindmapService::- addAssociation() Premi::Front-End::Services::MindmapService::- addNode() Premi::Front-End::Services::MindmapService::- deleteAssociation() Premi::Front-End::Services::MindmapService::- deleteNode() Premi::Front-End::Services::MindmapService::- updateNode() </pre>
TU81	<pre> Premi::Front-End::Services::MindmapService::- drawMap() Premi::Front-End::Services::MindmapService::- getNode() Premi::Front-End::Services::MindmapService::- listen() Premi::Front-End::Services::MindmapService::- stopListen() </pre>
TU82	<pre> Premi::Front-End::Services::PathService::- addNodeToPath() Premi::Front-End::Services::PathService::- addPath() Premi::Front-End::Services::PathService::- deletePath() Premi::Front-End::Services::PathService::- finalizePathUpdates() Premi::Front-End::Services::PathService::- getPath() </pre>

Test	Metodi
	<pre>Premi::Front-End::Services::PathService:- getPathNames() Premi::Front-End::Services::PathService:- removeNodeFromPath()</pre>
TU83	<pre>Premi::Front-End::Services::PresentationService:- getNodes() Premi::Front-End::Services::PresentationService:- getPath()</pre>
TU84	<pre>Premi::Front-End::Services::ProjectService:- createProject() Premi::Front-End::Services::ProjectService:- deleteProject() Premi::Front-End::Services::ProjectService:- finalizeProjectUpdates() Premi::Front-End::Services::ProjectService:- getCurrentProject() Premi::Front-End::Services::ProjectService:- getId() Premi::Front-End::Services::ProjectService:- getProjects() Premi::Front-End::Services::ProjectService:- getStyle() Premi::Front-End::Services::ProjectService:- loadProject()</pre>

Tabella 12: Tracciamento Test di Unità-Metodi

5.8 Tracciamento Metodi-Test di Unità

Metodo	Test
Premi::Back-End::App::Controllers::-ErrorHandler::handle()	TU5
Premi::Back-End::App::Controllers::Errors::-PremiError::PremiError()	TU8
Premi::Back-End::App::Controllers::Errors::-PremiError::toJSON()	TU8
Premi::Back-End::App::Controllers::Errors::-PremiError::toString()	TU8
Premi::Back-End::App::Controllers::-NotFoundHandler::handle()	TU6
Premi::Back-End::App::Controllers::Projects::-NodeController::addAssociation()	TU10
Premi::Back-End::App::Controllers::Projects::-NodeController::addNode()	TU11
Premi::Back-End::App::Controllers::Projects::-NodeController::associationById()	TU9
Premi::Back-End::App::Controllers::Projects::-NodeController::deleteAssociation()	TU10
Premi::Back-End::App::Controllers::Projects::-NodeController::deleteNode()	TU11
Premi::Back-End::App::Controllers::Projects::-NodeController::nodeById()	TU9
Premi::Back-End::App::Controllers::Projects::-NodeController::updateNode()	TU11
Premi::Back-End::App::Controllers::Projects::-PathController::addNodeToPath()	TU13
Premi::Back-End::App::Controllers::Projects::-PathController::addPath()	TU13
Premi::Back-End::App::Controllers::Projects::-PathController::deleteNodeFromPath()	TU13
Premi::Back-End::App::Controllers::Projects::-PathController::deletePath()	TU13
Premi::Back-End::App::Controllers::Projects::-PathController::getPath()	TU13
Premi::Back-End::App::Controllers::Projects::-PathController::pathById()	TU12
Premi::Back-End::App::Controllers::Projects::-PathController::updatePath()	TU13
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::addProject()	TU14
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::deleteProject()	TU14

Metodo	Test
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::getAllPaths()	TU14
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::getAllProjects()	TU14
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::getPresentation()	TU14
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::getProject()	TU14
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::projectById()	TU14
Premi::Back-End::App::Controllers::Projects::-ProjectManagementController::updateProject()	TU14
Premi::Back-End::App::Controllers::-StaticController::getUserManual()	TU7
Premi::Back-End::App::Controllers::Users::-AuthenticationController::signin()	TU15
Premi::Back-End::App::Controllers::Users::-AuthenticationController::signout()	TU15
Premi::Back-End::App::Controllers::Users::-AuthenticationController::signup()	TU15
Premi::Back-End::App::Controllers::Users::-AuthorizationController::requiresLogin()	TU16
Premi::Back-End::App::Models::NodeContentModel::-createNodeContent()	TU17
Premi::Back-End::App::Models::NodeModel::-getContents()	TU18
Premi::Back-End::App::Models::NodeModel::-updateContents()	TU18
Premi::Back-End::App::Models::NodeModel::-createNode()	TU18
Premi::Back-End::App::Models::PathModel::-addNode()	TU21
Premi::Back-End::App::Models::PathModel::-removeNode()	TU21
Premi::Back-End::App::Models::PathModel::-setName()	TU21
Premi::Back-End::App::Models::PathModel::-createPath()	TU19 TU20
Premi::Back-End::App::Models::ProjectModel::-addAssociation()	TU26
Premi::Back-End::App::Models::ProjectModel::-addNode()	TU24
Premi::Back-End::App::Models::ProjectModel::-addNodeToPath()	TU27

Metodo	Test
Premi::Back-End::App::Models::ProjectModel::- addPath()	TU27
Premi::Back-End::App::Models::ProjectModel::- getPathList()	TU28
Premi::Back-End::App::Models::ProjectModel::- getPathNodes()	TU28
Premi::Back-End::App::Models::ProjectModel::- removeAssociation()	TU26
Premi::Back-End::App::Models::ProjectModel::- removeNode()	TU25
Premi::Back-End::App::Models::ProjectModel::- removeNodeFromPath()	TU27
Premi::Back-End::App::Models::ProjectModel::- removePath()	TU27
Premi::Back-End::App::Models::ProjectModel::- setBkgColor()	TU23
Premi::Back-End::App::Models::ProjectModel::- setFontColor()	TU23
Premi::Back-End::App::Models::ProjectModel::- setFontFamily()	TU23
Premi::Back-End::App::Models::ProjectModel::- setName()	TU23
Premi::Back-End::App::Models::ProjectModel::- setPathName()	TU27
Premi::Back-End::App::Models::ProjectModel::- updateNode()	TU24
Premi::Back-End::App::Models::ProjectModel::- createProject()	TU22
Premi::Back-End::App::Models::RelationModel::- createRelation()	TU29
Premi::Back-End::App::Models::UserModel::- createUser()	TU30
Premi::Back-End::App::Routers::ProjectRouter::- router()	TU2
Premi::Back-End::App::Routers::StaticRouter::- router()	TU3
Premi::Back-End::App::Routers::UserRouter::- router()	TU4
Premi::Back-End::Server::start()	TU1
Premi::Front-End::AppConfig::- AppConfig()	TU31
Premi::Front-End::AppRouter::- AppRouter()	TU32
Premi::Front-End::AppRun::AppRun()	TU33

Metodo	Test
Premi::Front-End::Controllers::- AddToPathController::addButtonDisabled()	TU34
Premi::Front-End::Controllers::- AddToPathController::addClicked()	TU34
Premi::Front-End::Controllers::- AddToPathController::AddToPathController()	TU34
Premi::Front-End::Controllers::- AddToPathController::cancelClicked()	TU34
Premi::Front-End::Controllers::- AssociationAdderController::- AssociationAdderController()	TU35
Premi::Front-End::Controllers::- AssociationAdderController::cancelClicked()	TU35
Premi::Front-End::Controllers::- AssociationAdderController::confirmClicked()	TU35
Premi::Front-End::Controllers::- AssociationAdderController::confirmDisabled()	TU35
Premi::Front-End::Controllers::- ContextMenuController::call()	TU36
Premi::Front-End::Controllers::- ContextMenuController::cancel()	TU36
Premi::Front-End::Controllers::- ContextMenuController::ContextMenuController()	TU36
Premi::Front-End::Controllers::- DashboardController::DashboardController()	TU37
Premi::Front-End::Controllers::- DashboardController::deleteProjectClicked()	TU38
Premi::Front-End::Controllers::- DashboardController::editProjectClicked()	TU38
Premi::Front-End::Controllers::- DashboardController::newProjectClicked()	TU38
Premi::Front-End::Controllers::- DashboardController::startPresentationClicked()	TU38
Premi::Front-End::Controllers::- EditableNodeContentController::- EditableNodeContentController()	TU39
Premi::Front-End::Controllers::- EditableNodeContentController::selected()	TU39
Premi::Front-End::Controllers::- ErrorMessageController::ErrorMessageController()	TU40
Premi::Front-End::Controllers::- ErrorMessageController::hide()	TU40
Premi::Front-End::Controllers::HeaderController::- currentLocationContains()	TU42

Metodo	Test
Premi::Front-End::Controllers::HeaderController::fitMindmap()	TU42
Premi::Front-End::Controllers::HeaderController::getTitle()	TU42
Premi::Front-End::Controllers::HeaderController::HeaderController()	TU41
Premi::Front-End::Controllers::HeaderController::logout()	TU43
Premi::Front-End::Controllers::HeaderController::mapClicked()	TU43
Premi::Front-End::Controllers::HeaderController::pathClicked()	TU43
Premi::Front-End::Controllers::HeaderController::printPage()	TU43
Premi::Front-End::Controllers::HeaderController::setCurrentLocation()	TU43
Premi::Front-End::Controllers::HeaderController::showProjectSettingsEditor()	TU43
Premi::Front-End::Controllers::HeaderController::zoomMindmap()	TU42
Premi::Front-End::Controllers::HierarchicalMenuController::close()	TU44
Premi::Front-End::Controllers::HierarchicalMenuController::HierarchicalMenuController()	TU44
Premi::Front-End::Controllers::HierarchicalMenuController::isOpen()	TU44
Premi::Front-End::Controllers::HierarchicalMenuController::open()	TU44
Premi::Front-End::Controllers::LoginController::login()	TU45
Premi::Front-End::Controllers::LoginController::LoginController()	TU45
Premi::Front-End::Controllers::LoginController::signUp()	TU45
Premi::Front-End::Controllers::MindmapEditorController::addAssociationClicked()	TU47
Premi::Front-End::Controllers::MindmapEditorController::addNodeClicked()	TU47
Premi::Front-End::Controllers::MindmapEditorController::deleteNodeClicked()	TU47
Premi::Front-End::Controllers::MindmapEditorController::deleteRelationClicked()	TU47
Premi::Front-End::Controllers::MindmapEditorController::MindmapEditorController()	TU46

Metodo	Test
Premi::Front-End::Controllers::- MindmapEditorController::nodeSelected()	TU48
Premi::Front-End::Controllers::- MindmapEditorController::relationSelected()	TU48
Premi::Front-End::Controllers::- MindmapEditorController::resetNodeEdit()	TU49
Premi::Front-End::Controllers::- NodeContentsEditorController::addImageContent()	TU50
Premi::Front-End::Controllers::- NodeContentsEditorController::addTextContent()	TU50
Premi::Front-End::Controllers::- NodeContentsEditorController::annullaClicked()	TU50
Premi::Front-End::Controllers::- NodeContentsEditorController::deleteNodeContent()	TU50
Premi::Front-End::Controllers::- NodeContentsEditorController::- NodeContentsEditorController()	TU50
Premi::Front-End::Controllers::- NodeContentsEditorController::salvaClicked()	TU50
Premi::Front-End::Controllers::NodeController::- NodeController()	TU51
Premi::Front-End::Controllers::PathController::- PathController()	TU52
Premi::Front-End::Controllers::PathController::- removeNode()	TU52
Premi::Front-End::Controllers::PathController::- saveChanges()	TU52
Premi::Front-End::Controllers::- PathsEditorController::addNodeToPathClicked()	TU55
Premi::Front-End::Controllers::- PathsEditorController::close()	TU54
Premi::Front-End::Controllers::- PathsEditorController::deselectPath()	TU54
Premi::Front-End::Controllers::- PathsEditorController::destructor()	TU55
Premi::Front-End::Controllers::- PathsEditorController::getTongueName()	TU53
Premi::Front-End::Controllers::- PathsEditorController::isOpen()	TU53
Premi::Front-End::Controllers::- PathsEditorController::nodeDeselected()	TU55
Premi::Front-End::Controllers::- PathsEditorController::nodeSelected()	TU55
Premi::Front-End::Controllers::- PathsEditorController::open()	TU54

Metodo	Test
Premi::Front-End::Controllers::- PathsEditorController::PathsEditorController()	TU53
Premi::Front-End::Controllers::- PathsEditorController::selectPath()	TU54
Premi::Front-End::Controllers::- PathsListController::createNewPath()	TU56
Premi::Front-End::Controllers::- PathsListController::deletePath()	TU56
Premi::Front-End::Controllers::- PathsListController::pathSelected()	TU56
Premi::Front-End::Controllers::- PathsListController::PathsListController()	TU56
Premi::Front-End::Controllers::- PathsListController::present()	TU56
Premi::Front-End::Controllers::- PresentationController::classForNode()	TU57
Premi::Front-End::Controllers::- PresentationController::nextStep()	TU57
Premi::Front-End::Controllers::- PresentationController::PresentationController()	TU57
Premi::Front-End::Controllers::- PresentationController::previousStep()	TU57
Premi::Front-End::Controllers::- PresentationController::setCurrentStepFromId()	TU57
Premi::Front-End::Controllers::- PresentationController::setCurrentStepFromIndex()	TU57
Premi::Front-End::Controllers::- PresentationFooterController::leaveEvent()	TU58
Premi::Front-End::Controllers::- PresentationFooterController::- PresentationFooterController()	TU58
Premi::Front-End::Controllers::- PresentationFooterController::setFullscreen()	TU58
Premi::Front-End::Controllers::- PresentationFooterController::showFooter()	TU58
Premi::Front-End::Controllers::- PresentationFooterController::- stopWaitingForFooterHiding()	TU58
Premi::Front-End::Controllers::- PresentationViewerController::closeMenus()	TU61
Premi::Front-End::Controllers::- PresentationViewerController::isMenuOpen()	TU61
Premi::Front-End::Controllers::- PresentationViewerController::jumpToNode()	TU60

Metodo	Test
Premi::Front-End::Controllers::- PresentationViewerController::leaveMenu()	TU61
Premi::Front-End::Controllers::- PresentationViewerController::nextStep()	TU60
Premi::Front-End::Controllers::- PresentationViewerController::nodeChanged()	TU60
Premi::Front-End::Controllers::- PresentationViewerController::- PresentationViewerController()	TU59
Premi::Front-End::Controllers::- PresentationViewerController::previousStep()	TU60
Premi::Front-End::Controllers::- PresentationViewerController::printPresentation()	TU60
Premi::Front-End::Controllers::- PresentationViewerController::resumePresentation()	TU60
Premi::Front-End::Controllers::- PresentationViewerController::showMenus()	TU61
Premi::Front-End::Controllers::- PresentationViewerController::- stopWaitingForMenusHiding()	TU61
Premi::Front-End::Controllers::- ProjectSettingsEditorController::cancel()	TU62
Premi::Front-End::Controllers::- ProjectSettingsEditorController::confirm()	TU62
Premi::Front-End::Controllers::- ProjectSettingsEditorController::- ProjectSettingsEditorController()	TU62
Premi::Front-End::Controllers::- ProjectSettingsEditorController::selectChanged()	TU62
Premi::Front-End::Controllers::- ProjectsListController::ProjectsListController()	TU63
Premi::Front-End::Controllers::- ProjectsListController::showPaths()	TU63
Premi::Front-End::Controllers::- RegistrationController::createAccount()	TU64
Premi::Front-End::Controllers::- RegistrationController::logIn()	TU64
Premi::Front-End::Controllers::- RegistrationController::RegistrationController()	TU64
Premi::Front-End::Controllers::- SmartMenuController::close()	TU65
Premi::Front-End::Controllers::- SmartMenuController::isOpen()	TU65
Premi::Front-End::Controllers::- SmartMenuController::open()	TU65

Metodo	Test
Premi::Front-End::Controllers::- SmartMenuController::SmartMenuController()	TU65
Premi::Front-End::Model::ErrorInfo::- ErrorInfo()	TU66
Premi::Front-End::Model::ErrorInfo::- getCode()	TU66
Premi::Front-End::Model::ErrorInfo::- getMessage()	TU66
Premi::Front-End::Model::ErrorInfo::- getTitle()	TU66
Premi::Front-End::Model::Node::- addImage()	TU68
Premi::Front-End::Model::Node::- addText()	TU68
Premi::Front-End::Model::Node::- getContent()	TU68
Premi::Front-End::Model::Node::- getContents()	TU67
Premi::Front-End::Model::Node::- getTitle()	TU67
Premi::Front-End::Model::Node::- removeContent()	TU68
Premi::Front-End::Model::NodeContent::- getContent()	TU69
Premi::Front-End::Model::NodeContent::- getHeight()	TU69
Premi::Front-End::Model::NodeContent::- getId()	TU69
Premi::Front-End::Model::NodeContent::- getStyle()	TU70
Premi::Front-End::Model::NodeContent::- getType()	TU69
Premi::Front-End::Model::NodeContent::- getWidth()	TU69
Premi::Front-End::Model::NodeContent::- getX()	TU69
Premi::Front-End::Model::NodeContent::- getY()	TU69
Premi::Front-End::Model::NodeContent::- NodeContent()	TU69
Premi::Front-End::Model::NodeContent::- setHeight()	TU70
Premi::Front-End::Model::NodeContent::- setWidth()	TU70

Metodo	Test
Premi::Front-End::Model::NodeContent::- setX()	TU70
Premi::Front-End::Model::NodeContent::- setY()	TU70
Premi::Front-End::Model::NodeReference::- getId()	TU71
Premi::Front-End::Model::NodeReference::- getTitle()	TU71
Premi::Front-End::Model::Path::- addStep()	TU72
Premi::Front-End::Model::Path::- deleteStep()	TU72
Premi::Front-End::Model::Path::getId()	TU73
Premi::Front-End::Model::Path::- getName()	TU73
Premi::Front-End::Model::Path::- getSteps()	TU73
Premi::Front-End::Model::Path::- isDefault()	TU74
Premi::Front-End::Model::Path::- setName()	TU74
Premi::Front-End::Model::PresentationNode::- getAssociatedNodes()	TU75
Premi::Front-End::Model::PresentationNode::- getChildNodes()	TU75
Premi::Front-End::Model::PresentationNode::- getParentNode()	TU75
Premi::Front-End::Model::Project::- getBackgroundColor()	TU76
Premi::Front-End::Model::Project::- getFontFamily()	TU76
Premi::Front-End::Model::Project::- getId()	TU76
Premi::Front-End::Model::Project::- getName()	TU76
Premi::Front-End::Model::Project::- getRootId()	TU76
Premi::Front-End::Model::Project::- getTextColor()	TU76
Premi::Front-End::Model::Project::- setBackgroundColor()	TU77
Premi::Front-End::Model::Project::- setFontFamily()	TU77
Premi::Front-End::Model::Project::- setName()	TU77

Metodo	Test
Premi::Front-End::Model::Project::- setTextColor()	TU77
Premi::Front-End::Services::AuthenticationService::- login()	TU78
Premi::Front-End::Services::AuthenticationService::- logout()	TU78
Premi::Front-End::Services::AuthenticationService::- signUp()	TU78
Premi::Front-End::Services::MindmapAdapterService::- addAssociation()	TU79
Premi::Front-End::Services::MindmapAdapterService::- addNode()	TU79
Premi::Front-End::Services::MindmapAdapterService::- deleteAssociation()	TU79
Premi::Front-End::Services::MindmapAdapterService::- drawMap()	TU79
Premi::Front-End::Services::MindmapAdapterService::- getNode()	TU79
Premi::Front-End::Services::MindmapAdapterService::- listen()	TU79
Premi::Front-End::Services::MindmapAdapterService::- loadMap()	TU79
Premi::Front-End::Services::MindmapAdapterService::- setNode()	TU79
Premi::Front-End::Services::MindmapAdapterService::- stopListen()	TU79
Premi::Front-End::Services::MindmapService::- addAssociation()	TU80
Premi::Front-End::Services::MindmapService::- addNode()	TU80
Premi::Front-End::Services::MindmapService::- deleteAssociation()	TU80
Premi::Front-End::Services::MindmapService::- deleteNode()	TU80
Premi::Front-End::Services::MindmapService::- drawMap()	TU81
Premi::Front-End::Services::MindmapService::- getNode()	TU81
Premi::Front-End::Services::MindmapService::- listen()	TU81
Premi::Front-End::Services::MindmapService::- stopListen()	TU81
Premi::Front-End::Services::MindmapService::- updateNode()	TU80

Metodo	Test
Premi::Front-End::Services::PathService::- addNodeToPath()	TU82
Premi::Front-End::Services::PathService::- addPath()	TU82
Premi::Front-End::Services::PathService::- deletePath()	TU82
Premi::Front-End::Services::PathService::- finalizePathUpdates()	TU82
Premi::Front-End::Services::PathService::- getPath()	TU82
Premi::Front-End::Services::PathService::- getPathNames()	TU82
Premi::Front-End::Services::PathService::- removeNodeFromPath()	TU82
Premi::Front-End::Services::PresentationService::- getNodes()	TU83
Premi::Front-End::Services::PresentationService::- getPath()	TU83
Premi::Front-End::Services::ProjectService::- createProject()	TU84
Premi::Front-End::Services::ProjectService::- deleteProject()	TU84
Premi::Front-End::Services::ProjectService::- finalizeProjectUpdates()	TU84
Premi::Front-End::Services::ProjectService::- getCurrentProject()	TU84
Premi::Front-End::Services::ProjectService::- getId()	TU84
Premi::Front-End::Services::ProjectService::- getProjects()	TU84
Premi::Front-End::Services::ProjectService::- getStyle()	TU84
Premi::Front-End::Services::ProjectService::- loadProject()	TU84

Tabella 13: Tracciamento Metodi-Test di Unità

Appendice A Resoconto attività di verifica

In questa sezione del documento verranno descritti e commentati gli esiti delle attività di verifica svolte sui vari documenti prodotti durante le varie fasi del progetto.

A.1 Fase di Ammissione al progetto

Il tracciamento delle relazioni fra casi d'uso e requisiti, così come il tracciamento delle relazioni fra requisiti e fonti, è stato effettuato sfruttando l'applicativo *PragmaDB*, il quale facilita la verifica del tracciamento.

A.1.1 Analisi statica dei documenti

L'analisi dei documenti mediante *walkthrough_G* ha portato all'individuazione di alcuni errori frequenti a partire dai quali è stata stilata una lista di controllo da usare per le future attività di verifica.

Segue una lista degli errori riscontrati, ordinata dal più frequente al meno frequente.

- Mancato utilizzo dei comandi \LaTeX personalizzati;
- Mancato rispetto delle norme riguardanti gli elenchi puntati:
 - La prima parola di una voce dell'elenco non iniziava con una lettera maiuscola;
 - La voce dell'elenco terminava con un punto anziché con un punto e virgola (o viceversa).
- Errori riguardanti la struttura delle frasi: alcune frasi erano scritte in modo più colloquiale che formale o con tempi verbali errati;
- Errori di battitura.

A.1.1.1 Esiti verifiche automatizzate

Ogni documento è stato sottoposto anche a delle verifiche automatizzate per il calcolo dell'indice di Gulpease e per il controllo ortografico. Per maggiori informazioni si rimanda al processo 2.10.

Documento	Indice <i>Gulpease_G</i>	Esito
<i>Analisi dei Requisiti v1.0.0</i>	71,70	Superato
<i>Glossario v1.0.0</i>	43,87	Superato
<i>Norme di Progetto v1.0.0</i>	56,89	Superato
<i>Piano di Progetto v1.0.0</i>	51,03	Superato
<i>Piano di Qualifica v1.0.0</i>	52,57	Superato
<i>Studio di Fattibilità v1.0.0</i>	63,87	Superato
<i>II v1.0.0</i>	58,40	Superato

Tabella 14: Resoconto verifiche automatizzate - Fase di Ammissione al progetto

Dal calcolo dell'indice di Gulpease sono state escluse le tabelle e le didascalie.

A.1.2 Considerazioni finali

- Lo svolgimento delle attività di verifica è ancora poco automatizzato, questo perché senza una prima lista di errori comuni non è stato possibile predisporre automatismi per rilevarli e correggerli. Per le verifiche successive si cercherà di automatizzare ulteriormente questo processo;
- La stesura dei documenti ha raggiunto un buon grado di automatizzazione; per quanto riguarda l'*Analisi dei Requisiti* e il *Glossario*, infatti, la maggior parte del codice \LaTeX viene generato in modo automatico a partire dai dati presenti all'interno di *PragmaDB*. Non è stato possibile raggiungere lo stesso livello di automazione per gli altri documenti a causa della natura del loro contenuto.

A.2 Fase di Consolidamento dei requisiti

L'ampliamento di *PragmaDB* in vista della fase di progettazione ha permesso l'inserimento di nuovi script e funzionalità per il tracciamento e la verifica dei dati inseriti nei documenti. Un approccio così automatizzato alla stesura dei documenti ha permesso ai verificatori una maggiore facilità nell'individuazione degli errori e una maggiore facilità nell'individuazione delle singole componenti. In particolar modo si è cercato di facilitare l'analisi del tracciamento, che nella precedente fase è risultata particolarmente onerosa.

A.2.1 Analisi statica dei documenti

A.2.1.1 Esiti verifiche automatizzate

Ogni documento è stato sottoposto anche a delle verifiche automatizzate per il calcolo dell'indice di Gulpease e per il controllo ortografico. Per maggiori informazioni si rimanda al processo 2.10.

Documento	Indice <i>Gulpease_G</i>	Esito
<i>Analisi dei Requisiti v1.7.2</i>	67,82	Superato
<i>Glossario v1.5.2</i>	48,68	Superato
<i>Norme di Progetto v2.0.0</i>	58,06	Superato
<i>Piano di Progetto v1.2.2</i>	51,75	Superato
<i>Piano di Qualifica v1.15.2</i>	55,85	Superato
<i>E1 v1.0.0</i>	57,82	Superato
<i>E2 v1.0.0</i>	55,81	Superato

Tabella 15: Resoconto verifiche automatizzate - Fase di Consolidamento dei requisiti

Dal calcolo dell'indice di Gulpease sono state escluse le tabelle e le didascalie.

A.2.2 Considerazioni finali

L'attività di verifica ha raggiunto un buon grado di automatizzazione. Il processo di verifica dei documenti è stato migliorato su due punti ritenuti fondamentali:

- Una maggiore automatizzazione dei processi mediante l'aggiunta di nuove funzionalità a *PragmaDB* legate alla verifica del tracciamento tra i vari dati presenti nell'applicazione;

- La realizzazione di una lista di errori comuni che viene utilizzata per l'inspection dei documenti, che ha permesso di individuare in modo più efficace gli errori triviali commessi dai redattori dei documenti.

A.3 Fase di Progettazione dell'architettura

Per la fase di Progettazione dell'architettura si è deciso di ampliare le funzionalità di *PragmaDB*, affinché il processo di inserimento dei dati relativi a package, classi e il loro relativo tracciamento divenisse il più automatizzato possibile. Grazie a questi procedimenti le azioni di verifica e correzione sono risultate più semplici e immediate. Particolarmente utile è stato il sistema di verifica del tracciamento offerto da *PragmaDB*, che ha permesso di evitare la ricerca tramite *walkthrough_G* degli errori di tracciamento.

A.3.1 Analisi statica dei documenti

L'analisi dei documenti mediante *walkthrough_G* ha portato all'individuazione di alcuni errori frequenti a partire dai quali è stata stilata una lista di controllo da usare per le future attività di verifica.

Segue una lista degli errori riscontrati, ordinata dal più frequente al meno frequente:

- Mancato utilizzo dei comandi \LaTeX personalizzati;
- Mancato rispetto delle norme riguardanti gli elenchi puntati, specialmente quella riguardante l'inizio della voce con una lettera maiuscola;
- Alcune definizioni ricorrenti presenti nella descrizione dei componenti della *Specific Tecnica* sono state scritte in modo diverso;
- Errori riguardanti la struttura delle frasi: alcune frasi erano scritte in modo più colloquiale che formale o con tempi verbali errati.

A.3.1.1 Esiti verifiche automatizzate

Ogni documento è stato sottoposto anche a delle verifiche automatizzate per il calcolo dell'indice di Gulpease e per il controllo ortografico. Per maggiori informazioni si rimanda al processo 2.10.

Documento	Indice <i>Gulpease_G</i>	Esito
<i>Analisi dei Requisiti v1.11.2</i>	60,59	Superato
<i>Glossario v1.5.2</i>	48,68	Superato
<i>Norme di Progetto v2.0.0</i>	58,06	Superato
<i>Piano di Progetto v1.19.2</i>	52,12	Superato
<i>Piano di Qualifica v1.36.2</i>	56,21	Superato
<i>I2 v1.0.0</i>	59,80	Superato
<i>E3 v1.0.0</i>	57,53	Superato

Tabella 16: Resoconto verifiche automatizzate - Fase di Ammissione al progetto

Dal calcolo dell'indice di Gulpease sono state escluse le tabelle e le didascalie.

A.3.2 Considerazioni finali

- Il $team_G$ è soddisfatto della procedura di verifica attuale e del suo livello di automatizzazione.
- La stesura della *Specifica Tecnica* ha messo in rilievo alcune problematiche per quanto riguarda le definizioni ricorrenti, gli elenchi puntati e gli accenti di determinati caratteri. Si sta lavorando nella creazione di script che possano uniformare la documentazione e rendere aspetti di questo tipo il più automatici possibili.

A.4 Fase di Consolidamento dell'architettura

Visti gli ottimi risultati ottenuti durante la fase di **Progettazione dell'architettura** si è deciso di seguire un approccio simile nella stesura della *Definizione di Prodotto v2.0.0*.

A questo proposito si è deciso di ampliare le funzionalità di *PragmaDB* cercando di rendere il più automatico possibile l'inserimento all'interno della documentazione dei metodi e delle funzioni delle componenti definite nella **Progettazione dell'architettura**.

Questo ha permesso una maggiore facilità in fase di correzione del documento per quanto riguarda la verifica dei componenti inseriti e ha permesso una facile individuazione del tracciamento tra questi ultimi e i requisiti associati.

Sono stati poi aggiunti dei nuovi comandi \LaTeX per le descrizioni ricorrenti tra più componenti, in questo modo si è riusciti ad ottenere uno stile uniforme per il documento e a facilitare le attività di verifica.

Si è deciso in oltre di automatizzare la generazione delle tabelle per i vari test progettati. Per questo scopo *PragmaDB* è stato aggiornato con delle funzionalità in grado di creare automaticamente delle tabelle, volte a strutturare in maniera semplice ed elegante i test progettati con la loro descrizione ed il requisito o componente ad essi associato.

Questo ovviamente ha permesso, come per tutti i processi di automatizzazione, di avere una certa agevolazione durante il procedimento di verifica, in quanto molti errori di tracciamento o di formattazione del testo sono stati individuati in maniera automatica.

A.4.1 Analisi statica dei documenti

L'analisi dei documenti mediante $walkthrough_G$ ha portato all'individuazione di alcuni errori frequenti a partire dai quali è stata stilata una lista di controllo da usare per le future attività di verifica.

Segue una lista degli errori riscontrati, ordinata dal più frequente al meno frequente.

- Mancato rispetto delle norme riguardanti gli elenchi puntati:
 - La prima parola di una voce dell'elenco non iniziava con una lettera maiuscola;
 - La voce dell'elenco terminava con un punto anziché con un punto e virgola (o viceversa).
- Errori riguardanti la struttura delle frasi: alcune frasi erano scritte in modo più colloquiale che formale o con tempi verbali errati;
- Errori di battitura.

A.4.1.1 Esiti verifiche automatizzate

Ogni documento è stato sottoposto anche a delle verifiche automatizzate per il calcolo dell'indice di Gulpease e per il controllo ortografico. Per maggiori informazioni si rimanda al processo 2.10.

Documento	Indice <i>Gulpease_G</i>	Esito
<i>Analisi dei Requisiti v2.0.0</i>	60,45	Superato
<i>Glossario v2.0.0</i>	47,12	Superato
<i>Norme di Progetto v2.0.0</i>	58,06	Superato
<i>Piano di Progetto v2.0.0</i>	52,42	Superato
<i>Piano di Qualifica v2.0.0</i>	55,08	Superato
<i>Definizione di Prodotto v1.0.0</i>	60,01	Superato
<i>E4 v1.0.0</i>	69,90	Superato

Tabella 17: Resoconto verifiche automatizzate - Fase di Consolidamento dell'architettura

Dal calcolo dell'indice di Gulpease sono state escluse le tabelle e le didascalie.

A.4.2 Considerazioni finali

- Il *team_G* è molto soddisfatto del tempo speso nell'ampliamento di *PragmaDB* per quanto riguarda le procedure di stesura dei documenti. Infatti ciò ha facilitato la verifica e la correzione di essi;
- La stesura automatizzata dei test ha portato ancora una volta a degli errori per quanto riguarda la formattazione degli elenchi puntati. A tale proposito il *team_G* ha deciso di rivedere gli script associati e si è impegnato ad adottare una procedura più ferrea e precisa nella scrittura, in modo da ottenere dei documenti che siano omogenei nello stile di stesura.

A.5 Fase di Realizzazione del prodotto

Nel corso di questa fase il *team_G* ha puntato molto sulla misurazione dei livelli di qualità raggiunti e, a questo scopo, ha esteso *PragmaDB* con la funzionalità *Dashboard Metriche*, che ha permesso di automatizzare il calcolo di un grande numero di metriche e monitorare il loro soddisfacimento.

Questo miglioramento ha consentito di focalizzare l'attenzione sugli aspetti critici rilevati ed adottare delle strategie di correzione, al fine di far rientrare i valori delle metriche fra i parametri previsti.

Le migliorie apportate a *PragmaDB* nella fase precedente, inoltre, hanno permesso di semplificare molto il lavoro di stesura e verifica dei test di unità, in quanto è stato possibile definire nuovi comandi \LaTeX per le descrizioni ricorrenti ed individuare in maniera automatica molti errori di tracciamento o di formattazione del testo.

Grazie al lavoro svolto, inoltre, è stato possibile mantenere costantemente aggiornato il tracciamento degli esiti dei test che sono stati eseguiti e, parallelamente, sono stati automaticamente aggiornati i valori di tutte le metriche ad essi correlati.

A.5.1 Analisi statica dei documenti

La lista degli errori comuni individuati nelle fasi precedenti ha permesso di effettuare sui documenti un'analisi mediante *inspection_G* e portare all'individuazione di nuove occorrenze di anomalie rilevate nelle fasi precedenti, consentendo una loro rapida correzione.

Tale attività è stata comunque supportata anche da una parte di analisi mediante *walkthrough_G*, la quale ha portato all'individuazione di alcuni nuovi errori frequenti a partire dai quali è stata aggiornata la lista di controllo in previsione di future attività di verifica.

Segue una lista degli errori riscontrati, ordinata dal più frequente al meno frequente.

- Mancato rispetto delle norme riguardanti gli elenchi puntati:
 - La prima parola di una voce dell'elenco non iniziava con una lettera maiuscola;
 - La voce dell'elenco terminava con un punto anziché con un punto e virgola (o viceversa).
- Presenza di termini con lettere accentate errate o mancanti;
- Errori di battitura.

Il secondo punto di questo elenco merita particolare attenzione.

L'anomalia riguardante la *presenza di lettere accentate errate o mancanti* era stata individuata già nel corso delle attività di verifica effettuate in fasi precedenti, ed aveva portato alla formulazione di una strategia di correzione tale da risolvere la maggior parte delle occorrenze rilevate.

Le attività di verifica successive, però, non avevano fatto emergere che tale strategia risultava inefficace in alcuni documenti nei quali era stato usato un font \LaTeX particolare che, per motivi non chiari, permetteva l'inserimento di lettere accentate nel sorgente ma non consentiva una corretta resa di tali caratteri nel *pdf_G* prodotto.

A seguito di un'ulteriore segnalazione in sede di **Revisione di Progettazione**, i verificatori hanno quindi effettuato un controllo approfondito dei documenti prodotti, individuando concentrazioni sospettosamente maggiori di questa anomalia nei documenti che utilizzavano il font incriminato. A seguito di questa azione, si è deciso di cambiare il font per tali documenti, ed il problema è stato risolto con successo.

A.5.1.1 Esiti verifiche automatizzate

Ogni documento è stato sottoposto anche a delle verifiche automatizzate per il calcolo dell'indice di Gulpease e per il controllo ortografico. Per maggiori informazioni si rimanda al processo 2.10.

Documento	Indice <i>Gulpease_G</i>	Esito
<i>Analisi dei Requisiti v2.0.0</i>	60,45	Superato
<i>Glossario v2.0.0</i>	47,12	Superato
<i>Norme di Progetto v3.0.0</i>	57.66	Superato
<i>Piano di Progetto v3.0.0</i>	51.09	Superato
<i>Piano di Qualifica v3.0.0</i>	54.10	Superato
<i>Definizione di Prodotto v2.0.0</i>	67.19	Superato
<i>Manuale Utente v1.0.0</i>	46.18	Superato
<i>Revisione di Bilancio v1.0.0</i>	54.36	Superato
<i>E5 v1.0.0</i>	52.89	Superato
<i>E6 v1.0.0</i>	75.45	Superato

Tabella 18: Resoconto verifiche automatizzate - Fase di Realizzazione del prodotto

Dal calcolo dell'indice di Gulpease sono state escluse le tabelle e le didascalie.

A.5.2 Soddisfaccimento metriche

Al fine di monitorare costantemente il livelli di qualità raggiunti, grande attenzione è stata posta nella misurazione, secondo le metriche identificate, di processi (sezione 2) e prodotto (sezione 3). In particolare, per automatizzare il più possibile il calcolo di questi valori, la *piattaforma_G PragmaDB* è stata arricchita della funzionalità *Dashboard Metriche*, in grado di mantenere aggiornati i valori delle principali metriche nel corso dell'avanzamento dei lavori e di verificare il rispetto degli intervalli di accettabilità ed ottimalità fissati.

Metrica	Unità di misura	Valore	Accettazione	Ottimalità
Disponibilità <i>PragmaDB</i>	<i>Percentuale</i>	95.22	80 – 100	90 – 100
Tempo correzione incoerenze <i>PragmaDB</i>	<i>Giorni</i>	2.18	0 – 3	0 – 1
Errori individuazione termini glossario	<i>Termini</i>	2.74	0 – 3	0
Schedule Variance	<i>Attività</i>	0	≥ 0	≥ 0
Budget Variance	<i>Euro</i>	55.00	≥ 0	≥ 0
Rischi non preventivati	<i>Rischi</i>	1	0 – 5	0
Efficienza gestione rischi	<i>Giorni</i>	18.42	≥ 20	≥ 60
SFIN - Ottimalità	<i>Percentuale</i>	41.38	≥ 30	≥ 50
SFOUT - Non Accettabilità	<i>Percentuale</i>	5.75	0 – 6	0 – 3
Metodi per classe - Non Accettabilità	<i>Percentuale</i>	12.64	0 – 15	0 – 5
Parametri per metodo - Non Accettabilità	<i>Percentuale</i>	0	0 – 5	0 – 3
Produttività di codifica	<i>Media</i>	14.43	≥ 3	≥ 10
Complessità Ciclomantica - Non Accettabilità	<i>Percentuale</i>	0	0 – 1	0
Linee commento su linee codice	<i>Percentuale</i>	34.93	≥ 25	≥ 30

Metrica	Unità di misura	Valore	Accettazione	Ottimalità
Variabili inutilizzate	Percentuale	0	0	0
Halstead Difficulty per-function - Non Accettabilità	Percentuale	0	0 – 1	0
Indice di manutenibilità - Non Accettabilità	Percentuale	5.26	0 – 10	0 – 5
Copertura dei test	Percentuale	92	70 – 100	80 – 100

Tabella 19: Metriche principali

Criticità principali rilevate:

- **Efficienza di gestione dei rischi**

- **Problema:** il valore calcolato per questa metrica risulta particolarmente basso rispetto alle aspettative in seguito al manifestarsi di situazioni problematiche a causa della gestione non abbastanza efficace di due rischi:

- * *Problemi legati ad un errore di bilancio:* il rischio non era stato previsto nell'attività di *analisi dei rischi* effettuata in precedenza, e questo ha portato tale problematica a trasformarsi in un problema serio in tempo molto breve rispetto alla sua individuazione, a causa della mancanza di una strategia precedentemente pianificata;

- * *Problemi tra i componenti del gruppo:* il rischio era già stato preventivato nell'attività di *analisi dei rischi* effettuata in precedenza ed erano già state applicate delle strategie al fine di mitigarlo; le soluzioni adottate, però, non sono risultate pienamente soddisfacenti, costringendo il *team_G* a utilizzare un approccio differente.

- **Strategie:** si rimanda al documento *Piano di Progetto v3.0.0*.

A.5.3 Considerazioni finali

- L'incremento di attenzione nei confronti delle metriche di qualità definite è stata molto proficua, in quanto ha permesso l'individuazione di criticità non note ed ha consentito di mettere in pratica alcune strategie volte ad elevare i livelli di qualità raggiunti;
- Le funzionalità di tracciamento implementate nelle fasi successive in *PragmaDB* si sono effettivamente rivelate fondamentali per mantenere costantemente aggiornati tutti i dati, in particolare quelli riguardanti requisiti, metriche e test definiti.
- L'attività di verifica ha sottolineato nuovamente delle problematiche inerenti alla formattazione degli elenchi puntati; in particolar modo è stato preso atto della grande difficoltà di definire delle norme tipografiche univoche che siano in grado di coprire tutti i diversi casi di utilizzo di tali elementi. Per questo motivo, il *team_G* si è proposto di rivedere nelle fasi successive le norme correlate, al fine di permettere una maggiore copertura dei casi d'utilizzo emersi.