

Carbon12 – Predire in Grafana

Piano di Qualifica

Informazioni sul documento

Versione	1.4.0
Stato	Approvato
Data di creazione	2019/11/29
Data di approvazione	2020/05/11
Redazione	Nicolò Fassina Francesco Gobbo Andrea Longo Alessandro Lovo Veronica Pederiva
Verifica	Giacomo Callegari Manuel De Franceschi Andrea Longo
Approvazione	Andrea Longo
Uso	Esterno
Destinatari	Carbon12 Zucchetti SPA Prof. Tullio Vardanega Prof. Riccardo Cardin
E-mail di riferimento	carbon.dodici@gmail.com

Scopo del documento

Definizione delle operazioni di test, verifica e validazione eseguite dal gruppo Carbon12 durante la realizzazione del progetto Predire in Grafana.

Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.4.0-0	2020/05/11	Approvazione del documento	Andrea Longo	Responsabile
b.3.1-0	2020/05/11	Superamento verifica	Giacomo Callegari	Verificatore
b.3.0-3	2020/05/10	Aggiornamento Appendice A	Francesco Gobbo	Verificatore
b.3.0-2	2020/05/10	Redazione Appendice B4 – Valutazioni conclusive	Nicolò Fassina	Verificatore
b.3.0-1	2020/05/09	Aggiornamento sezione §2.2 – Test implementati e soddisfatti	Francesco Gobbo	Verificatore
b.3.0-0	2020/04/13	Approvazione del documento	Francesco Gobbo	Responsabile
b.2.1-0	2020/04/13	Validazione del documento	Andrea Longo	Verificatore
b.2.0-11	2020/04/12	Verifica del documento	Giacomo Callegari	Verificatore
		Redazione Appendice B3	Veronica Pederiva	Verificatore
b.2.0-10	2020/04/12	Aggiornamento Appendice A	Francesco Gobbo	Verificatore
b.2.0-9	2020/04/08	Verifica del documento	Manuel De Franceschi	Verificatore
		Aggiornamento Appendice A	Francesco Gobbo	Verificatore
a.2.0-8	2020/04/06	Aggiornamento sezione §2.2.7 Tabella 6 - Test d'Unità Plug-in	Andrea Longo	Progettista
a.2.0-7	2020/04/05	Aggiornamento sezione §2.2.6 Tabella 5 - Test di Integrazione Plug-in	Andrea Longo	Progettista
a.2.0-6	2020/04/01	Verifica del documento	Giacomo Callegari	Verificatore
		Aggiornamento Appendice A	Francesco Gobbo	Verificatore
a.2.0-5	2020/03/31	Redazione sezione §2.2.6 Tabella 5 - Test di Integrazione Applicazione di Addestramento	Francesco Gobbo	Progettista
a.2.0-4	2020/03/29	Redazione sezione §2.2.7 Tabella 6 - Test d'Unità Applicazione di Addestramento	Alessandro Lovo	Progettista
a.2.0-3	2020/03/26	Verifica del documento	Manuel De Franceschi	Verificatore
		Aggiornamento Appendice A	Francesco Gobbo	Verificatore
a.2.0-2	2020/03/25	Aggiornamento riferimenti requisiti sezione §2.2.4 Tabella 2, sezione §2.2.5 Tabella 3 e Tabella 4	Andrea Longo	Progettista
a.2.0-1	2020/03/24	Correzione struttura documento	Alessandro Lovo	Progettista
a.2.0-0	2020/03/08	Approvazione del documento	Nicolò Fassina	Responsabile
a.1.1-0	2020/03/08	Validazione del documento	Giacomo Callegari	Verificatore
a.1.0-6	2020/03/08	Verifica del documento	Andrea Longo	Verificatore
a.1.0-6	2020/03/07	Aggiornamento Appendice B	Veronica Pederiva	Verificatore

Versione	Data	Descrizione	Nominativo	Ruolo
a.1.0-5	2020/03/07	Aggiornamento Appendice A	Francesco Gobbo	Verificatore
a.1.0-5	2020/02/29	Aggiornamento Appendice A	Francesco Gobbo	Verificatore
a.1.0-4	2020/02/18	Aggiornamento Appendice A	Francesco Gobbo	Verificatore
0.1.0-3	2020/02/13	Aggiornamento sezione §2.7.4, §2.7.5	Francesco Gobbo	Verificatore
0.1.0-2	2020/02/09	Verifica del documento	Nicolò Fassina	Verificatore
0.1.0-2	2020/02/08	Aggiornamento Appendice A	Francesco Gobbo	Verificatore
0.1.0-1	2020/02/01	Riordino struttura documento; aggiunta sezione §2.7	Francesco Gobbo	Verificatore
0.0.1-0	2020/01/13	Approvazione del documento	Giacomo Callegari	Responsabile
0.0.0-13	2020/01/12	Verifica del documento	Andrea Longo	Verificatore
0.0.0-13	2020/01/11	Redazione Appendice B	Nicolò Fassina	Verificatore
0.0.0-12	2020/01/10	Correzioni e integrazioni al documento	Francesco Gobbo	Amministratore
0.0.0-11	2020/01/09	Revisione del documento	Manuel De Franceschi	Verificatore
0.0.0-11	2020/01/08	Redazione Appendice C	Francesco Gobbo	Amministratore
0.0.0-10	2020/01/04	Correzioni alle metriche	Alessandro Lovo	Amministratore
0.0.0-9	2020/01/04	Revisione del documento	Manuel De Franceschi	Verificatore
0.0.0-9	2020/01/03	Redazione della sezione 3.3 – Processi Organizzativi	Alessandro Lovo	Amministratore
0.0.0-8	2020/01/02	Redazione della sezione 3.1 – Processi Primari, 3.2 – Processi di Supporto	Alessandro Lovo	Amministratore
0.0.0-7	2019/12/30	Redazione della sezione 2.4 – Usabilità, 2.5 – Manutenibilità, 2.6 - Portabilità	Nicolò Fassina	Amministratore
0.0.0-6	2019/12/29	Redazione della sezione 2.1 – Funzionalità, 2.2 – Affidabilità, 2.3 – Efficienza	Nicolò Fassina	Amministratore
0.0.0-5	2019/12/26	Revisione struttura	Nicolò Fassina	Amministratore
0.0.0-4	2019/12/24	Redazione Appendice A	Francesco Gobbo	Amministratore
0.0.0-3	2019/12/22	Redazione capitolo 1 – Introduzione	Francesco Gobbo	Amministratore
0.0.0-2	2019/12/04	Redazione struttura documento	Francesco Gobbo	Amministratore
0.0.0-1	2019/11/29	Creazione del documento	Andrea Longo	Responsabile

Indice

1 Introduzione	1
1.1 Scopo del documento	1
1.2 Scopo del prodotto	1
1.3 Documenti complementari	1
1.4 Riferimenti	1
1.4.1 Normativi	1
1.4.2 Informativi	1
2 Qualità del software	3
2.1 Obiettivi delle metriche	3
2.2 Test	4
2.2.1 Introduzione	4
2.2.2 Modello a V	4
2.2.3 Copertura dei Test	5
2.2.3.1 Criteri di Copertura	5
2.2.4 Test d'accettazione	6
2.2.5 Test di sistema	11
2.2.5.1 Tracciamento test di sistema – requisito funzionale	13
2.2.6 Test d'integrazione	15
2.2.7 Test d'unità	19
2.2.8 Mock	31
3 Qualità di processo	32
3.1 Obiettivi delle metriche	32
A Resoconto attività di verifica	34
A.1 Analisi statica dei documenti	34
A.2 Esiti verifiche della qualità di Prodotto	34
A.2.1 Funzionalità	34
A.2.2 Affidabilità	35
A.2.3 Efficienza	35
A.2.4 Usabilità	35
A.2.5 Manutenibilità	36
A.2.6 Portabilità	37
A.3 Esiti verifiche della qualità di Processo	37
A.3.1 Analisi dei Requisiti	37
A.3.2 Progettazione Architettuale	37
A.3.3 Progettazione di dettaglio	38
A.3.4 Pianificazione – Schedule Variance (SV)	38
A.3.5 Pianificazione – Budget Variance (BV)	39
A.3.5.1 Periodo di Analisi	39
A.3.5.2 Periodo di Progettazione della Base Tecnologica	39
A.3.5.3 Periodo di Progettazione di dettaglio e codifica	40
A.3.5.4 Periodo di Validazione e collaudo	40
A.3.5.5 Budget Variance Totale	41
A.3.6 Documentazione	42
A.3.6.1 Indice di Gulpase (IG)	42
A.3.6.1 Correttezza Ortografica (IG)	46
A.3.7 Verifica	46
A.3.7 Copertura dei Test	46
A.3.7 Gestione Qualità	47
B Valutazioni per il miglioramento	48

B.1 Fase di Analisi.....	48
B.1.1 Valutazione sull'organizzazione.....	48
B.1.2 Valutazione sui ruoli.....	49
B.1.2.1 Amministratore.....	49
B.1.2.2 Responsabile.....	49
B.1.2.3 Analista.....	49
B.1.2.4 Verificatore.....	50
B.1.3 Valutazione sull'uso degli strumenti.....	50
B.2 Fase di Progettazione della base tecnologica	50
B.2.1 Valutazione sull'organizzazione.....	50
B.2.2 Valutazione sui ruoli.....	51
B.2.2.1 Amministratore.....	51
B.2.2.2 Responsabile.....	51
B.2.2.3 Analista.....	51
B.2.2.4 Progettisti.....	51
B.2.2.5 Programmatori.....	52
B.2.2.6 Verificatori.....	52
B.2.3 Valutazione sull'uso degli strumenti.....	52
B.3 Periodo di Progettazione di dettaglio e codifica	52
B.3.1 Valutazione sull'organizzazione.....	52
B.3.2 Valutazione sui ruoli.....	52
B.3.3 Valutazione sull'uso degli strumenti.....	53
B.4 Periodo di Valutazione e collaudo.....	53
B.4.1 Valutazione conclusiva.....	53

1 Introduzione

Il *Piano di Qualifica* è un documento che prevede una lavorazione continua per l'intero sviluppo del progetto, poiché molti dei contenuti trattati sono considerati di natura instabile.

Viene prodotto per costruzione e non per correzione, limitando così lo spreco di risorse. La costruzione applicata è di tipo incrementale e i contenuti iniziali sono da considerarsi incompleti, in quanto le parti non presenti verranno aggiornate successivamente.

1.1 Scopo del documento

Il *Piano di Qualifica* ha l'obiettivo di definire le strategie adottate per la verifica e la validazione, preservando la qualità di prodotto e di processo. Durante le attività viene applicato un sistema di verifica continua al fine di conseguire lo scopo prefissato. Operando in questo modo è possibile rilevare e attivarsi per la correzione delle irregolarità riscontrate, mantenendo l'EFFICIENZAG e l'EFFICACIAG.

1.2 Scopo del prodotto

Lo scopo del prodotto è l'ottenimento di un PLUG-ING applicabile in GRAFANAG, che monitori il flusso di dati fornendo un'analisi predittiva su di essi basata sull'ADDESTRAMENTOG di modelli di SUPPORT-VECTOR MACHINEG e REGRESSIONE LINEAREG. I dati monitorati e le predizioni misurate saranno visualizzati su una Dashboard contenente i grafici prodotti dal sistema di creazione di grafici di Grafana.

1.3 Documenti complementari

Onde evitare possibili situazioni di ambiguità nella lettura del documento, viene fornito il *Glossario v.1.4.0*. I termini o concetti che si sono ritenuti rilevanti e che necessitano di una ulteriore specifica sono contrassegnati con la dicitura G a pedice e sono evidenziati in maiuscoletto.

1.4 Riferimenti

1.4.1 Normativi

- Capitolato d'Appalto C4 – *Predire in Grafana*
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C4.pdf>
- Norme di Progetto: *Norme di progetto v.1.4.0*

1.4.2 Informativi

- Piano di Progetto: *Piano di Progetto v.1.4.0*
- ISO/IEC 9126
https://en.wikipedia.org/wiki/ISO/IEC_9126
- Standard ISO/IEC 12207:1995 sezione 5, 6, 7 pp. 9-47:
https://www.math.unipd.it/~tullio/IS-1/2009/Appfondimenti/ISO_12207-1995.pdf
- ISO/IEC 15504
https://en.wikipedia.org/wiki/ISO/IEC_15504
- Ciclo di Deming (PDCA)
https://it.wikipedia.org/wiki/Ciclo_di_Deming

- HM&S - SPICE Process Assessment Model
<http://www.spice121.com/cms/en/about-spice-1-2-1.html>
- Indice di Gulpease
https://it.wikipedia.org/wiki/Indice_Gulpease
- Schedule Variance e metriche correlate:
<https://www.smartsheet.com/hacking-pmp-how-calculate-schedule-variance>
- Slide corso “Ingegneria del Software” - UNIPD: Qualità di Processo e di Prodotto
<https://www.math.unipd.it/~tullio/IS-1/2019/Dispense/L12.pdf>
<https://www.math.unipd.it/~tullio/IS-1/2019/Dispense/L13.pdf>
- HintSW
<https://www.hintsw.com/it/safety-engineering/pianificazione-e-concezione-del-sw/modello-a-v-di-sviluppo-del-sw.html>

2 Qualità del software

Per garantire la qualità del prodotto software, il gruppo Carbon12 ha deciso di far riferimento al modello di qualità descritto nello standard *ISO/IEC 9126*. In questa sezione verranno trattate le caratteristiche che incidono sulla capacità del prodotto software di soddisfare i requisiti impliciti ed espliciti riportando le metriche utilizzate per valutare la qualità ricercata.

2.1 Obiettivi delle metriche

Tipo qualità	Nome metriche	Intervallo accettabile	Intervallo desiderabile
Funzionalità	Funzionalità sviluppate (FS)	[60%, 100%]	[70%, 100%]
	Correttezza funzionale (CF)	[80%, 100%]	[90%, 100%]
Affidabilità	Densità degli errori (DE)	[0%, 15%]	[0%, 5%]
Efficienza	Tempo medio di risposta (TMR)	[0s, 5s]	[0s, 3s]
	Consumo medio di potenza elaborativa (CMPE)	[0%, 30%]	[0%, 15%]
	Consumo medio di memoria (CMM)	[0MB, 500MB]	[0MB, 100MB]
Usabilità	Ambiguità delle funzioni (AF)	[0%, 5%]	[0%, 2%]
	Ambiguità degli errori (AE)	[0%, 10%]	[0%, 5%]
	Descrizione delle funzioni (DF)	[80%, 100%]	[90%, 100%]
	Tempo medio di apprendimento (TMA)	3h	2h
	Estetica dell'interfaccia (EI)	[7, 10]	[8, 10]
Manutenibilità	Presenza di commenti (PC)	[10%, 30%]	[10%, 20%]
	Complessità ciclomatica (CC)	[1, 10]	[1, 7]
	Impatto negativo delle modifiche (INM)	[0%, 10%]	0%
	Completamento dei test (CT)	[80%, 100%]	[90%, 100%]
Portabilità	Browser supportati (BS)	{(Chrome, 32), (Firefox, 27)}	{(Chrome, 32), (Firefox, 27), (Opera, 19), (Internet Explorer, 10)}

TABELLA 1 – OBIETTIVI DELLE METRICHE PER LA QUALITÀ DEL SOFTWARE

2.2 Test

2.2.1 Introduzione

Un'attività fondamentale della Verifica è il controllo. Esistono due tipi di controlli: controlli statici e controlli dinamici. Mentre i controlli statici sono basati sulla non esecuzione del programma, ad esempio analisi tramite INSPECTIONG e WALKTHROUGHG, i controlli dinamici richiedono l'esecuzione del programma. Questi ultimi sono chiamati anche Test, intesi sia come attività del processo di sviluppo sia come singola estensione di prova di un programma identificata dall'insieme di condizioni (input e condizioni di esecuzione) in cui essa si svolge.

I test hanno come obiettivo principale la verifica della correttezza funzionale di un programma o di un sistema, ma anche la realizzazione di controlli mirati alla valutazione di fattori di qualità, come: affidabilità, usabilità ed efficienza.

In questa sezione del *Piano di Qualifica* vengono presentati i test e la qualità che apportano allo sviluppo del prodotto, secondo la regolamentazione indicata nella sezione §3.5.4.2 delle *Norme di Progetto*.

2.2.2 Modello a V

Il MODELLO A V_G è il modello da noi adottato per l'implementazione dei Test, sviluppati in parallelo allo sviluppo del prodotto software durante le attività di analisi e progettazione. Questo modello ci permetterà di adottare un metodo di lavoro che verifichi sia le parti del prodotto sviluppate, sia di progettare le specifiche del prodotto in modo tale che siano coerenti con gli obiettivi da raggiungere. Dal modello si può intuire che opereremo con un approccio top/down integrando i vari livelli dei Test durante la fase di progettazione, per poi svilupparlo con la codifica partendo dai Test d'unità e raggiungere la soddisfazione con i Test d'Accettazione.

Le caratteristiche di questo modello sono le seguenti:

- **Frecce grandi:** il flusso di lavoro della costruzione è lineare;
- **Frecce orizzontali:** si mette in evidenza la relazione tra i vari livelli di test e le diverse descrizioni del sistema;
 - I Test d'Accettazione si progettano tenendo conto dei requisiti utente e vengono utilizzati per il collaudo del prodotto;
 - I Test di Sistema si basano sulle funzionalità offerte dal sistema completo rispetto ai requisiti software;
 - I Test d'Integrazione vengono guidati dall'architettura software;
 - I Test d'Unità si basano sui risultati della progettazione in dettaglio;
- **Frecce diagonali:** rappresentano l'influenza che la progettazione dei test di ciascun livello può avere sull'attività di progettazione del livello successivo.

L'utilizzo di questo modello permette di progettare i test molto prima dell'effettiva esecuzione degli stessi. Ciò che non viene mostrato in questo schema sono i probabili cicli di rielaborazione conseguenti all'esecuzione dei test.

2.2.4 Test d'accettazione

Codice	Descrizione	Stato	Esito
TAFO1	L'Amministratore deve poter allenare il sistema in una applicazione web apposita. All'Amministratore viene chiesto di: 1. Accedere alla pagina per l'addestramento.	NI	NS
TAFO1.1	L'Amministratore può accedere alla pagina web per la generazione del predittore ed inserire i dati per l'addestramento. All'Amministratore viene chiesto di: 1. Accedere alla pagina web; 1.1. Inserire un file CSV con i dati di addestramento; 1.2. Scegliere se inserire un file JSON con il predittore già allenato 1.3. Scegliere la tipologia di modello desiderata; 1.4. Scegliere se inserire una funzione da applicare ai dati; 1.5. Scaricare il predittore in formato JSON.	NI	NS
TAFO1.1.1	L'Amministratore può inserire un file contenente i dati per l'addestramento. All'Amministratore viene chiesto di: 1. Attivare un apposito tasto, per il caricamento del file per l'addestramento. 2. Ricercare e Selezionare il file da caricare; 3. Confermare la scelta del file di dati da caricare	NI	NS
TAFO1.1.1.3	L'applicativo comunica, tramite un messaggio, il fallimento del caricamento del file. Al sistema viene chiesto di: 1. Verificare se il file caricato non dovesse essere idoneo per l'addestramento e notificare l'amministratore con un messaggio d'errore; 2. Scartare il file caricato; 3. Riportare l'Amministratore alla pagina iniziale.	NI	NS
TAFO1.1.2	L'Amministratore può inserire un file contenente il vecchio predittore. All'Amministratore viene chiesto di: 1. Attivare un apposito tasto, per il caricamento del vecchio file predittore. 2. Ricercare e Selezionare il file da caricare; 3. Confermare la scelta del file di dati da caricare	NI	NS

TABELLA 2 – TEST D'ACCETTAZIONE

Codice	Descrizione	Stato	Esito
TAF01.1.2.3	<p>L'applicativo comunica, tramite un messaggio, il fallimento del caricamento del file.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Verificare se il file caricato non dovesse essere idoneo per l'addestramento e notificare l'amministratore con un messaggio d'errore; 2. Scartare il file caricato; 3. Riportare l'Amministratore alla pagina iniziale. 	NI	NS
TAF01.1.3	<p>L'Amministratore può scegliere la tipologia del modello desiderata.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare l'opzione per selezionare la tipologia di modello desiderata; 2. Visionare la lista dei modelli disponibili; 3. Selezionare il modello che desidera addestrare. 4. Confermare la scelta del modello. 	NI	NS
TAFF1.1.4	<p>L'Amministratore sceglie di inserire una funzione da applicare al flusso dei dati.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare l'opzione (pulsante) dedicata all'inserimento di una funzione; 2. Visionare la lista delle funzioni disponibili; 3. Selezionare la funzione che si desidera inserire; 4. Confermare la scelta della funzione. 	NI	NS
TAF01.1.5	<p>L'applicativo (pagina web, autorizzata dal sistema) offre all'Amministratore una modalità per esportare i dati generati.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare un pulsante per l'esportazione del file generato; 2. Visualizzare nei download il file contenente i dati generati scaricato dal sistema. 	NI	NS
TAFF1.2	<p>L'Amministratore desidera attivare l'addestramento continuo.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante dedicato all'addestramento continuo; 2. Visionare i modelli disponibili; 3. Selezionare il modello da applicare; 4. Visualizzare la conferma dal sistema dell'avvio dell'addestramento continuo. 	NI	NS

TABELLA 2 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TAFF1.3	<p>Il sistema mette a disposizione dell'Amministratore una modalità di interruzione dell'addestramento continuo.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante per interrompere l'addestramento continuo; 2. Visualizzare la conferma dal sistema dell'interruzione dell'addestramento continuo. 	NI	NS
TAFF1.4	<p>Il sistema mette a disposizione dell'Amministratore un metodo per l'addestramento direttamente in Grafana.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante dedicato all'avvio dell'addestramento; 2. Selezionare la sorgente dati per l'estrazione dati; 3. Scegliere il modello desiderato; 4. Confermare le scelte effettuate. 	NI	NS
TAFO2	<p>Il sistema permette all'Amministratore di configurare il plug-in.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Caricare il file ottenuto dall'addestramento; 2. Selezionare i nodi su cui effettuare la predizione; 3. Scegliere la tipologia di visualizzazione della previsione. 	NI	NS
TAFO2.1	<p>Il sistema offre all'Amministratore una modalità per caricare il file con i risultati dell'addestramento.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante per il caricamento del file. 2. Cercare e selezionare il file che si vuole caricare; 4. Confermare che il file caricato corrisponde a quello desiderato; 5. Visualizzare la notifica di corretto caricamento dal sistema. 	NI	NS
TAFD2.1.3.1	<p>Il sistema notifica all'Amministratore il fallimento del caricamento del file dell'addestramento.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Inviare l'errore in cui è segnalato che è stato caricato un file non valido; 2. Reindirizzare l'Amministratore nella scelta del file. 	NI	NS

TABELLA 2 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TAFD2.1.3.2	<p>Il sistema notifica all'Amministratore che non è stato caricato alcun file.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Inviare l'errore in cui è segnalato che non è stato caricato alcun file; 2. Reindirizzare l'Amministratore nella scelta del file. 	NI	NS
TAFO2.2	<p>Il sistema offre all'Amministratore un metodo per selezionare i nodi su cui effettuare la predizione.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Visionare la lista dei nodi caricati dal file del modello addestrato; 2. Selezionare i nodi desiderati; 3. Confermare la scelta dei nodi selezionati; 4. Visualizzare la notifica di conferma della selezione dei nodi. 	NI	NS
TAFD2.2.3.1	<p>Il sistema notifica all'Amministratore che è stato selezionato un nodo non idoneo.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Inviare la notifica di errore in cui è segnalata la selezione di un nodo NON idoneo. 2. Reindirizzare l'Amministratore nella selezione dei nodi. 	NI	NS
TAFD2.2.3.2	<p>Il sistema notifica all'Amministratore che non è stato selezionato alcun nodo.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Inviare la notifica di errore in cui è segnalato che non è stato selezionato alcun nodo. 2. Reindirizzare l'Amministratore nella selezione dei nodi. 	NI	NS
TAFO2.3	<p>Il sistema offre all'Amministratore un metodo per la selezione del tipo di visualizzazione della predizione.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Visionare l'elenco delle tipologie di visualizzazione disponibili; 2. Selezionare il tipo di visualizzazione della predizione; 3. Confermare la scelta della visualizzazione desiderata. 4. Visualizzare la conferma dal sistema che l'operazione è andata a buon fine. 	NI	NS

TABELLA 2 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TAFD2.3.3.1	<p>Il sistema notifica all'Amministratore che non è stato selezionato alcun tipo di visualizzazione.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Inviare la notifica di errore di mancata selezione della tipologia di visualizzazione; 2. Reindirizzare l'Amministratore nella selezione della tipologia di visualizzazione. 	NI	NS
TAFO3	<p>Il sistema permette all'Utente di avviare la predizione.</p> <p>All'Utente viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante per avviare la predizione; 2. Visualizzare la conferma che la predizione è stata avviata. 	NI	NS
TAFO4	<p>Il sistema permette all'Utente di terminare la predizione.</p> <p>All'Utente viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante per terminare la predizione; 2. Visualizzare la conferma che la predizione è stata interrotta. 	NI	NS
TAFD5	<p>Il sistema permette all'Amministratore di impostare degli alert.</p> <p>All'Amministratore viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante per l'inserimento di alert; 2. Impostare la soglia desiderata dell'alert; 3. Confermare le caratteristiche dell'alert; 4. Visualizzare la notifica di conferma della corretta impostazione dell'alert. 	NI	NS
TAFD6	<p>Il sistema notifica all'Amministratore un messaggio di errore nel caso in cui si stia cercando di eliminare un pannello con la previsione attiva.</p> <p>Al sistema viene chiesto di:</p> <ol style="list-style-type: none"> 1. Inviare la notifica di errore di impossibilità di eliminazione di un pannello, a causa della previsione attiva; 2. Reindirizzare l'Amministratore nella visualizzazione dei pannelli. 	NI	NS
TAFO7	<p>Il sistema offre all'Utente un metodo per la visualizzazione delle previsioni.</p> <p>All'Utente viene chiesto di:</p> <ol style="list-style-type: none"> 1. Attivare il pulsante che permette di visualizzare le previsioni e i relativi dettagli. 	NI	NS

TABELLA 2 CONTINUAZIONE

2.2.5 Test di sistema

Codice	Descrizione	Stato	Esito
TS1	Viene verificato che il sistema permetta all'Amministratore di allenare il sistema.	I	S
TS1.1	Viene verificato che la pagina web dedicata all'addestramento sia raggiungibile dall'Amministratore.	I	S
TS1.1.1	Viene verificato che il sistema renda attivabile il tasto per il caricamento del file per l'addestramento all'Amministratore.	I	S
TS1.1.1.1	Viene verificato che il sistema permetta di selezionare il file per l'addestramento all'Amministratore.	I	S
TS1.1.1.2	Viene verificato che il sistema permetta la conferma del file selezionato all'Amministratore.	I	S
TS1.1.1.3	Viene verificato che il sistema comunichi, tramite un messaggio, il fallimento del caricamento del file.	I	S
TS1.1.2	Viene verificato che il sistema renda attivabile il tasto per il caricamento del file contenente il precedente predittore all'Amministratore.	I	S
TS1.1.2.1	Viene verificato che il sistema permetta di selezionare il file contenente il precedente predittore all'Amministratore.	I	S
TS1.1.2.2	Viene verificato che il sistema permetta la conferma del file selezionato all'Amministratore.	I	S
TS1.1.2.3	Viene verificato che il sistema comunichi, tramite un messaggio, il fallimento del caricamento del file.	I	S
TS1.1.3	Viene verificato che il sistema permetta la scelta della tipologia di modello desiderato all'Amministratore.	I	S
TS1.1.3.1	Viene verificato che il sistema permetta la selezione del modello "Regressione Lineare" all'Amministratore.	I	S
TS1.1.3.2	Viene verificato che il sistema permetta la selezione del modello "Support Vector Machine" all'Amministratore.	I	S
TS1.1.3.3	Viene verificato che il sistema permetta la selezione del modello "Reti Neurali" all'Amministratore.	NI	NS
TS1.1.4	Viene verificato che il sistema permetta di inserire una funzione da applicare al flusso dei dati all'Amministratore.	NI	NS
TS1.1.5	Viene verificato che il sistema renda disponibile un metodo per esportare i dati generati all'Amministratore.	I	S
TS1.2	Viene verificato che il sistema permetta l'attivazione dell'addestramento continuo all'Amministratore.	NI	NS
TS1.3	Viene verificato che il sistema renda disponibile un metodo per interrompere l'addestramento continuo all'Amministratore.	NI	NS
TS1.4	Viene verificato che il sistema renda disponibile un metodo per l'addestramento direttamente in Grafana all'Amministratore.	NI	NS

TABELLA 3 – TEST DI SISTEMA

Codice	Descrizione	Stato	Esito
TS2	Viene verificato che il sistema permetta la configurazione del plug-in all'Amministratore.	I	S
TS2.1	Viene verificato che il sistema permetta il caricamento del modello addestrato all'Amministratore.	I	S
TS2.1.1	Viene verificato che il sistema renda disponibile un metodo per caricare il file all'Amministratore.	I	S
TS2.1.2	Viene verificato che il sistema permetta la selezione del file desiderato all'Amministratore.	I	S
TS2.1.3	Viene verificato che il sistema richieda all'Amministratore che il file caricato è quello destinato all'attività di analisi predittiva.	I	S
TS2.1.3.1	Viene verificato che il sistema notifichi all'Amministratore il fallimento del caricamento del file dell'addestramento.	I	S
TS2.1.3.2	Viene verificato che il sistema notifichi all'Amministratore il mancato caricamento del file dell'addestramento.	I	S
TS2.2	Viene verificato che il sistema renda disponibile un metodo per selezionare i nodi su cui effettuare la predizione all'Amministratore.	I	S
TS2.2.1	Viene verificato che il sistema permetta la visione della lista dei nodi caricati dal file di addestramento all'Amministratore.	I	S
TS2.2.2	Viene verificato che il sistema permetta la selezione dei nodi su cui effettuare la predizione all'Amministratore.	I	S
TS2.2.3	Viene verificato che il sistema richieda la conferma della scelta dei nodi all'Amministratore.	I	S
TS2.2.3.1	Viene verificato che il sistema notifichi all'Amministratore che è stato selezionato un nodo non idoneo.	I	S
TS2.2.3.2	Viene verificato che il sistema notifichi all'Amministratore che non è stato selezionato alcun nodo.	I	S
TS2.2.4	Viene verificato che il sistema notifichi all'Amministratore che la selezione dei nodi è andata a buon fine.	I	S
TS2.3	Viene verificato che il sistema renda disponibile un metodo per la selezione del tipo di visualizzazione della predizione all'Amministratore.	I	S
TS2.3.1	Viene verificato che il sistema permetta la selezione del tipo di visualizzazione all'Amministratore.	I	S
TS2.3.2	Viene verificato che il sistema permetta la selezione del tipo di visualizzazione tra indicatore o grafico all'Amministratore.	I	S
TS2.3.3	Viene verificato che il sistema richieda all'Amministratore di confermare il tipo di visualizzazione.	I	S
TS2.3.3.1	Viene verificato che il sistema notifichi all'Amministratore che non è stato selezionato alcun tipo di visualizzazione.	I	S
TS2.3.4	Viene verificato che il sistema notifichi all'Amministratore che la selezione è andata a buon fine.	I	S

TABELLA 3 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TS3	Viene verificato che il sistema permetta l'avvio della predizione all'Utente.	I	S
TS4	Viene verificato che il sistema permetta di terminare la predizione all'Utente.	I	S
TS5	Viene verificato che il sistema permetta l'impostazione degli alert all'Amministratore.	I	S
TS5.1	Viene verificato che il sistema renda disponibile un metodo per avviare l'aggiunta di un alert all'Amministratore.	I	S
TS5.2	Viene verificato che il sistema renda disponibile un metodo per l'impostazione della soglia dell'alert all'Amministratore.	I	S
TS5.3	Viene verificato che il sistema richieda la conferma dell'alert all'Amministratore.	I	S
TS5.4	Viene verificato che il sistema notifichi l'Amministratore quando la soglia viene impostata correttamente.	I	S
TS6	Viene verificato che il sistema notifichi all'Amministratore un messaggio di errore nel caso in cui si stia cercando di eliminare un pannello con la previsione ancora attiva.	NI	NS
TS7	Viene verificato che il sistema renda disponibile un metodo per la visualizzazione delle previsioni e dei relativi dettagli all'Utente.	I	S

TABELLA 3 CONTINUAZIONE

2.2.5.1 Tracciamento test di sistema – requisito funzionale

Test	Requisito
TS1	RFO1
TS1.1	RFO1.1
TS1.1.1	RFO1.1.1
TS1.1.1.1	RFO1.1.1.1
TS1.1.1.2	RFO1.1.1.2
TS1.1.1.3	RFD1.1.1.3
TS1.1.2	RFO1.1.2
TS1.1.2.1	RFO1.1.2.1
TS1.1.2.2	RFO1.1.2.2
TS1.1.2.3	RFO1.1.2.3
TS1.1.3	RFO1.1.3
TS1.1.3.1	RFO1.1.3.1
TS1.1.3.2	RFO1.1.3.2
TS1.1.3.3	RFF1.1.3.3
TS1.1.4	RFF1.1.4
TS1.1.5	RFO1.1.5
TS1.2	RFF1.2

TABELLA 4 - TRACCIAMENTO TEST DI SISTEMA - REQUISITO FUNZIONALE

Test	Requisito
TS1.3	RFF1.3
TS1.4	RFF1.4
TS2	RFO2
TS2.1	RFO2.1
TS2.1.1	RFO2.1.1
TS2.1.2	RFO2.1.2
TS2.1.3	RFO2.1.3
TS2.1.3.1	RFD2.1.3.1
TS2.1.3.2	RFD2.1.3.2
TS2.2	RFO2.2
TS2.2.1	RFO2.2.1
TS2.2.2	RFO2.2.2
TS2.2.3	RFO2.2.3
TS2.2.3.1	RFD2.2.3.1
TS2.2.3.2	RFD2.2.3.2
TS2.2.4	RFD2.2.4
TS2.3	RFO2.3
TS2.3.1	RFO2.3.1
TS2.3.2	RFO2.3.2
TS2.3.3	RFO2.3.3
TS2.3.3.1	RFD2.3.3.1
TS2.3.4	RFD2.3.4
TS3	RFO3
TS4	RFO4
TS5	RFO5
TS5.1	RFD5.1
TS5.2	RFD5.2
TS5.3	RFD5.3
TS5.4	RFD5.4
TS6	RFD6
TS7	RFO7

TABELLA 4 CONTINUAZIONE

2.2.6 Test d'integrazione

Codice	Descrizione	Stato	Esito
Addestramento			
TI1	Viene verificato che il server venga avviato correttamente. (Server)	I	S
TI2	Viene verificato che il metodo savePredittore salvi correttamente il file nel disco. (Server)	I	S
TI3	Viene verificato che la funzione validityCSV invii un messaggio di errore se la struttura del file.csv non dovesse essere corretta. (Server)	I	S
TI4	Viene verificato che la funzione validityCSV invii una stringa vuota nel caso la struttura del file.csv dovesse essere corretta. (Server)	I	S
TI5	Viene verificato che la funzione validityJSON invii un messaggio di errore se la struttura del file.json non dovesse essere corretta perché il titolo non corrisponde a quello aspettato. (Server)	I	S
TI6	Viene verificato che la funzione validityJSON invii un messaggio di errore se la struttura del file.json non dovesse essere corretta perché le keys non corrispondono a quelle aspettate. (Server)	I	S
TI7	Viene verificato che la funzione validityJSON invii un messaggio di errore se la versione del file addestramento non dovesse essere compatibile. (Server)	I	S
TI8	Viene verificato che la funzione validityJSON invii un messaggio di errore se le sorgenti del file.json non corrispondono a quelle contenute nel file.csv, perché sono meno di quelle che ci si aspetta. (Server)	I	S
TI9	Viene verificato che la funzione validityJSON invii un messaggio di errore se le sorgenti del file.json non corrispondono a quelle contenute nel file.csv, perché non corrispondono a quelle aspettate. (Server)	I	S
TI10	Viene verificato che la funzione validityJSON invii un messaggio di errore se il modello selezionato nel file.json non corrisponde a quello che si vuole allenare. (Server)	I	S
TI11	Viene verificato che la funzione validityJSON invii una stringa vuota nel caso non si trovassero errori. (Server)	I	S
TI12	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse SVM, ma non si caricasse l'ex-predittore. (Server)	I	S
TI13	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse SVM e si caricasse l'ex-predittore. (Server)	I	S
TI14	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse RL, ma non si caricasse l'ex-predittore. (Server)	I	S
TI15	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse RL e si caricasse l'ex-predittore. (Server)	I	S
TI16	Viene verificato che il metodo autoGetColumns ritorni il vettore con le intestazioni colonne del file.csv. (CsvReader)	I	S
TI17	Viene verificato che il metodo labelsColumn venga settata la colonna corrispondente ai labels. (CsvReader)	I	S
TI18	Viene verificato che il metodo getData ritorni un'array nullo. (CsvReader)	I	S

TABELLA 5 - TEST D'INTEGRAZIONE

Codice	Descrizione	Stato	Esito
TI19	Viene verificato che il metodo getData ritorni l'array {M*N}. (CsvReader)	I	S
TI20	Viene verificato che il metodo autoGetData legga correttamente i dati presenti nel file.csv. (CsvReader)	I	S
TI21	Viene verificato che il metodo autoGetLabel legga correttamente le labels presenti nel file.csv. (CsvReader)	I	S
TI22	Viene verificato che il metodo getDataSource legga correttamente le sources presenti nel file.csv. (CsvReader)	I	S
TI23	Viene verificato che il metodo countSource conti correttamente il numero delle sources. (CsvReader)	I	S
TI24	Viene verificato che il metodo checkVersion verifichi la compatibilità della versione del plug-in e dell'addestramento. (RPredittore)	I	S
TI25	Viene verificato che il metodo validity testi se il file.json ha le corrette keys e il titolo corretto. (RPredittore)	I	S
TI26	Viene verificato che il metodo getTitle ritorni il titolo che ci si aspetta del file.json. (RPredittore)	I	S
TI27	Viene verificato che il metodo getPluginVersion ritorni la versione che ci si aspetta. (RPredittore)	I	S
TI28	Viene verificato che il metodo getTrainVersion ritorni la versione che ci si aspetta. (RPredittore)	I	S
TI29	Viene verificato che il metodo getDataEntry ritorni l'array delle dataEntry che ci si aspetta. (RPredittore)	I	S
TI30	Viene verificato che il metodo getDataEntry ritorni un array vuoto. (RPredittore)	I	S
TI31	Viene verificato che il metodo getModel ritorni il modello che ci si aspetta. (RPredittore)	I	S
TI32	Viene verificato che il metodo getFileVersion ritorni la versione del numero di volte di addestramento aspettato. (RPredittore)	I	S
TI33	Viene verificato che il metodo getNotes ritorni le note che ci si aspetta. (RPredittore)	I	S
TI34	Viene verificato che il metodo getConfiguration ritorni la configurazione che ci si aspetta. (RPredittore)	I	S
TI35	Viene verificato che il metodo fromJSON ritorni la configurazione del JSON che ci si aspetta. (RIAdapter)	I	S
TI36	Viene verificato che il metodo trainJSON ritorni la configurazione del predittore che ci si aspetta dopo l'allenamento. (RIAdapter)	I	S
TI37	Viene verificato che il metodo fromJSON ritorni la configurazione del JSON che ci si aspetta. (SvmAdapter)	I	S
TI38	Viene verificato che il metodo trainJSON ritorni la configurazione del predittore che ci si aspetta dopo l'allenamento. (SvmAdapter)	I	S
TI39	Viene verificato che il metodo setHeader setti l'header che ci si aspetta. (WPredittore)	I	S
TI40	Viene verificato che il metodo setHeader setti l'header che ci si aspetta senza passagli title. (WPredittore)	I	S

TABELLA 5 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TI41	Viene verificato che il metodo setDataEntry setti correttamente le data_entry. (WPredittore)	I	S
TI42	Viene verificato che il metodo setModel setti correttamente il modello. (WPredittore)	I	S
TI43	Viene verificato che il metodo setFileVersion setti correttamente la versione di addestramento. (WPredittore)	I	S
TI44	Viene verificato che il metodo setNotes setti correttamente le note. (WPredittore)	I	S
TI45	Viene verificato che il metodo setConfiguration setti correttamente la configurazione. (WPredittore)	I	S
TI46	Viene verificato che il metodo save ritorni il JSON che ci si aspetta. (WPredittore)	I	S
Plug-in			
TI66	Viene verificato che Predict venga costruito correttamente. (PredictCtrl)	I	S
TI67	Viene verificato che il metodo verifyDashboard ritorni una dashboard corretta (PredictCtrl)	I	S
TI68	Viene verificato che il metodo verifyDashboard ritorna false se non esistesse la dashboard. (PredictCtrl)	I	S
TI80	Viene verificato che il metodo buildTarget costruisca correttamente un Target. (Builder)	I	S
TI81	Viene verificato che il metodo buildView costruisca correttamente un View quando il modello è SVM. (Builder)	I	S
TI82	Viene verificato che il metodo buildView costruisca correttamente un View quando il modello è RL. (Builder)	I	S
TI83	Viene verificato che il metodo addPanel aggiunga correttamente il pannello. (Dashboard)	I	S
TI84	Viene verificato che il metodo storeSettings salvi correttamente la configurazione della dashboard. (Dashboard)	I	S
TI85	Viene verificato che Influx venga costruito correttamente. (Influx)	I	S
TI86	Viene verificato che il metodo getLastValue ritorni un array contenente l'ultimo valore memorizzato nel database. (Influx)	I	S
TI87	Viene verificato che il metodo deleteAllPrediction elimini correttamente tutte le predizioni. (Influx)	I	S
TI88	Viene verificato che il metodo getJSON ritorni correttamente la rappresentazione del pannello in formato JSON. (Panel)	I	S
TI89	Viene verificato che il metodo setBackendSrv imposti correttamente la configurazione della predizione. (PredictLooper)	I	S
TI90	Viene verificato che il metodo setConfig imposti correttamente la configurazione. (PredictLooper)	I	S
TI91	Viene verificato che il metodo setConfig imposti correttamente la prima configurazione. (PredictLooper)	I	S
TI92	Viene verificato che il metodo setInflux imposti correttamente il riferimento al database. (PredictLooper)	I	S

TABELLA 5 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TI93	Viene verificato che il metodo dbWrite scriva correttamente il risultato della predizione sul database. (PredictLooper)	I	S
TI94	Viene verificato che il metodo startPrediction avvii correttamente la predizione nel caso in cui essa sia stata definita. (PredictLooper)	I	S
TI95	Viene verificato che il metodo startPrediction non avvii correttamente la predizione nel caso in cui essa non sia stata definita. (PredictLooper)	I	S
TI96	Viene verificato che il metodo stopPrediction fermi la predizione. (PredictLooper)	I	S
TI97	Viene verificato che il metodo getPrediction ritorni la predizione che ci si aspetta nel caso il modello fosse SVM. (PredictLooper)	I	S
TI98	Viene verificato che il metodo getPrediction ritorni la predizione che ci si aspetta nel caso il modello fosse RL. (PredictLooper)	I	S
TI99	Viene verificato che il metodo predictSVM predica il valore che ci si aspetta. (PredictLooper)	I	S
TI100	Viene verificato che il metodo predictRL predica il valore che ci si aspetta. (PredictLooper)	I	S
TI101	Viene verificato che predict ritorni la predizione che ci si aspetta. (RIAdapter)	I	S
TI102	Viene verificato che predictClass ritorni la predizione che ci si aspetta. (SvmAdapter)	I	S

TABELLA 5 TEST CONTINUAZIONE

2.2.7 Test d'unità

Codice	Descrizione	Stato	Esito
Addestramento			
TU1	Viene verificato che durante la costruzione del server si verifichi il caso in cui il termini per l'errore dovuto alla mancanza del file config.json. (Server)	I	S
TU2	Viene verificato che durante la costruzione del server non si verifichino errori. (Server)	I	S
TU3	Viene verificato che il server venga avviato correttamente. (Server)	I	S
TU4	Viene verificato che la funzione config funzioni correttamente definendo la configurazione. (Server)	I	S
TU5	Viene verificato che la funzione getCSVColumns funzioni correttamente fornendo le intestazioni delle colonne. (Server)	I	S
TU6	Viene verificato che la funzione getChartData funzioni correttamente fornendo i dati contenuti nel file.csv. (Server)	I	S
TU7	Viene verificato che la funzione downloadPredittore funzioni correttamente permettendo di scaricare il file. (Server)	I	S
TU8	Viene verificato che nella funzione uploadForm ritorni la response desiderata se viene caricato il file.json, se viene passato il nome del futuro predittore con l'estensione.json e il file.csv non genera un errore e il file.json non genera errore. (Server)	I	S
TU9	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se viene caricato il file.json, se viene passato il nome del futuro predittore con l'estensione.json e il file.csv non genera un errore, ma il file.json genera errore. (Server)	I	S
TU10	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se viene caricato il file.json, se viene passato il nome del futuro predittore senza l'estensione.json e il file.csv non genera un errore, ma il file.json genera errore. (Server)	I	S
TU11	Viene verificato che nella funzione uploadForm ritorni la response desiderata se viene caricato il file.json, se viene passato il nome del futuro predittore senza l'estensione.json e il file.csv non genera un errore e il file.json non genera errore. (Server)	I	S
TU12	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se viene caricato il file.json, se non viene passato il nome del futuro predittore senza l'estensione.json, ma il file.csv genera un errore. (Server)	I	S
TU13	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se viene caricato il file.json, se viene passato il nome del futuro predittore con l'estensione.json, ma il file.csv genera un errore. (Server)	I	S
TU14	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se viene caricato il file.json, se viene passato il nome del futuro predittore senza l'estensione.json, ma il file.csv genera un errore. (Server)	I	S

TABELLA 6 - TEST D'UNITÀ

Codice	Descrizione	Stato	Esito
TU15	Viene verificato che nella funzione uploadForm ritorni la response desiderata se non viene caricato il file.json, se viene passato il nome del futuro predittore con l'estensione.json e il file.csv non genera un errore. (Server)	I	S
TU16	Viene verificato che nella funzione uploadForm ritorni la response desiderata se non viene caricato il file.json, se viene passato il nome del futuro predittore senza l'estensione.json, e il file.csv genera un errore. (Server)	I	S
TU17	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se viene caricato il file.json, se non viene passato il nome del futuro predittore senza l'estensione.json e il file.csv non genera un errore, ma il file.json genera errori. (Server)	I	S
TU18	Viene verificato che nella funzione uploadForm non ritorni la response desiderata se non viene caricato il file.json, se non viene passato il nome del futuro predittore senza l'estensione.json, il file.csv non genera un errore. (Server)	I	S
TU19	Viene verificato che nella funzione uploadForm ritorni la response desiderata se viene caricato il file.json, se non viene passato il nome del futuro predittore senza l'estensione.json, il file.csv non genera un errore e il file.json non genera errore. (Server)	I	S
TU20	Viene verificato che il metodo savePredittore salvi correttamente il file nel disco. (Server)	I	S
TU21	Viene verificato che la funzione validityCSV invii un messaggio di errore se la struttura del file.csv non dovesse essere corretta. (Server)	I	S
TU22	Viene verificato che la funzione validityCSV invii una stringa vuota nel caso la struttura del file.csv dovesse essere corretta. (Server)	I	S
TU23	Viene verificato che la funzione validityJSON invii un messaggio di errore se la struttura del file.json non dovesse essere corretta perché il titolo non corrisponde a quello aspettato. (Server)	I	S
TU24	Viene verificato che la funzione validityJSON invii un messaggio di errore se la struttura del file.json non dovesse essere corretta perché le keys non corrispondono a quelle aspettate. (Server)	I	S
TU25	Viene verificato che la funzione validityJSON invii un messaggio di errore se la versione del file addestramento non dovesse essere compatibile. (Server)	I	S
TU26	Viene verificato che la funzione validityJSON invii un messaggio di errore se le sorgenti del file.json non corrispondono a quelle contenute nel file.csv, perché sono meno di quelle che ci si aspetta. (Server)	I	S
TU27	Viene verificato che la funzione validityJSON invii un messaggio di errore se le sorgenti del file.json non corrispondono a quelle contenute nel file.csv, perché non corrispondono a quelle aspettate. (Server)	I	S
TU28	Viene verificato che la funzione validityJSON invii un messaggio di errore se il modello selezionato nel file.json non corrisponde a quello che si vuole allenare. (Server)	I	S
TU29	Viene verificato che la funzione validityJSON invii una stringa vuota nel caso non si trovassero errori. (Server)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU30	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse SVM, ma non si caricasse l'ex-predittore. (Server)	I	S
TU31	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse SVM e si caricasse l'ex-predittore. (Server)	I	S
TU32	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse RL, ma non si caricasse l'ex-predittore. (Server)	I	S
TU33	Viene verificato che la funzione train funzioni correttamente nel caso in cui si scegliesse RL e si caricasse l'ex-predittore. (Server)	I	S
TU34	Viene verificata la costruzione di csv_reader con path e data e option. (CsvReader)	I	S
TU35	Viene verificata la costruzione di csv_reader con path e data, ma senza option. (CsvReader)	I	S
TU36	Viene verificato la costruzione di csv_reader, ma il file.csv è vuoto. (CsvReader)	I	S
TU37	Viene verificato che il metodo autoGetColumns ritorni il vettore con le intestazioni delle colonne del file.csv. (CsvReader)	I	S
TU38	Viene verificato che il metodo labelsColumn venga settata la colonna corrispondente ai labels. (CsvReader)	I	S
TU39	Viene verificato che il metodo getData ritorni un'array nullo. (CsvReader)	I	S
TU40	Viene verificato che il metodo getData ritorni l'array {1*N}. (CsvReader)	I	S
TU41	Viene verificato che il metodo getData ritorni l'array {N*M}. (CsvReader)	I	S
TU42	Viene verificato che il metodo autoGetData legga correttamente i dati presenti nel file.csv. (CsvReader)	I	S
TU43	Viene verificato che il metodo autoGetLabel legga correttamente le labels presenti nel file.csv. (CsvReader)	I	S
TU44	Viene verificato che il metodo getDataSource legga correttamente le sources presenti nel file.csv. (CsvReader)	I	S
TU45	Viene verificato che il metodo countSource conti correttamente il numero delle sources. (CsvReader)	I	S
TU46	Viene verificato che autoGetColumns sia implementata correttamente. (DataReader)	I	S
TU47	Viene verificato che setLabelsColumn sia implementata correttamente. (DataReader)	I	S
TU48	Viene verificato che autoGetData sia implementata correttamente. (DataReader)	I	S
TU49	Viene verificato che autoGetLabel sia implementata correttamente. (DataReader)	I	S
TU50	Viene verificato che getDataSource sia implementata correttamente. (DataReader)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU51	Viene verificato che fromJSON sia implementata correttamente. (Model)	I	S
TU52	Viene verificato che train sia implementata correttamente. (Model)	I	S
TU53	Viene verificata la costruzione di RPredditore senza avere dati. (RPredditore)	I	S
TU54	Viene verificata la costruzione di RPredditore con dati. (RPredditore)	I	S
TU55	Viene verificato che il metodo checkVersion verifichi la compatibilità della versione del plug-in e dell'addestramento. (RPredditore)	I	S
TU56	Viene verificato che il metodo versionToInt torni il valore numerico della versione di plug-in e addestramento. (RPredditore)	I	S
TU57	Viene verificato che il metodo versionToInt non ritorni un valore numerico se la versione che gli viene passata non dovesse rispettare le aspettative. (RPredditore)	I	S
TU58	Viene verificato che il metodo validity testi se il file.json ha le corrette keys e il titolo corretto. (RPredditore)	I	S
TU59	Viene verificato che il metodo getTitle ritorni il titolo che ci si aspetta del file.json. (RPredditore)	I	S
TU60	Viene verificato che il metodo getPluginVersion ritorni la versione che ci si aspetta. (RPredditore)	I	S
TU61	Viene verificato che il metodo getTrainVersion ritorni la versione che ci si aspetta. (RPredditore)	I	S
TU62	Viene verificato che il metodo getDataEntry ritorni l'array delle dataEntry che ci si aspetta. (RPredditore)	I	S
TU63	Viene verificato che il metodo getDataEntry ritorni un array vuoto. (RPredditore)	I	S
TU64	Viene verificato che il metodo getModel ritorni il modello che ci si aspetta. (RPredditore)	I	S
TU65	Viene verificato che il metodo getFileVersion ritorni la versione del numero di volte di addestramento aspettato, nel caso in cui sia stato allenato solo una volta. (RPredditore)	I	S
TU66	Viene verificato che il metodo getFileVersion ritorni la versione del numero di volte di addestramento aspettato. (RPredditore)	I	S
TU67	Viene verificato che il metodo getNotes ritorni le note che ci si aspetta. (RPredditore)	I	S
TU68	Viene verificato che il metodo getConfiguration ritorni la configurazione che ci si aspetta. (RPredditore)	I	S
TU69	Viene verificato che la costruzione di RL_Adapter avvenga in modo corretto. (RLAdapter)	I	S
TU70	Viene verificato che il metodo fromJSON ritorni la configurazione del JSON che ci si aspetta. (RLAdapter)	I	S
TU71	Viene verificato che il metodo trainJSON ritorni la configurazione del predittore che ci si aspetta dopo l'allenamento. (RLAdapter)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU72	Viene verificato che la costruzione di SVM_Adapter avvenga in modo corretto. (SvmAdapter)	I	S
TU73	Viene verificato che il metodo fromJSON ritorni la configurazione del JSON che ci si aspetta. (SvmAdapter)	I	S
TU74	Viene verificato che il metodo trainJSON ritorni la configurazione del predittore che ci si aspetta dopo l'allenamento. (SvmAdapter)	I	S
TU75	Viene verificato che la costruzione di WPredittore avvenga in modo corretto. (WPredittore)	I	S
TU76	Viene verificato che il metodo setHeader setti l'header che ci si aspetta. (WPredittore)	I	S
TU77	Viene verificato che il metodo setHeader setti l'header che ci si aspetta senza passagli title. (WPredittore)	I	S
TU78	Viene verificato che il metodo setDataEntry setti correttamente le data_entry. (WPredittore)	I	S
TU79	Viene verificato che il metodo setModel setti correttamente il modello. (WPredittore)	I	S
TU80	Viene verificato che il metodo setFileVersion setti correttamente la versione di addestramento. (WPredittore)	I	S
TU81	Viene verificato che il metodo setNotes setti correttamente le note. (WPredittore)	I	S
TU82	Viene verificato che il metodo setConfiguration setti correttamente la configurazione. (WPredittore)	I	S
TU83	Viene verificato che il metodo save ritorni il JSON che ci si aspetta. (WPredittore)	I	S
Plug-in			
TU84	Viene verificato che ImportCtrl venga costruito correttamente. (ImportCtrl)	I	S
TU85	Viene verificato che il metodo onUpload carichi il file del predittore correttamente quando il file è valido ed il modello è SVM. (ImportCtrl)	I	S
TU86	Viene verificato che il metodo onUpload carichi il file del predittore correttamente quando il file è valido e il modello non è SVM. (ImportCtrl)	I	S
TU87	Viene verificato che il metodo onUpload ritorni il modello aspettato. (ImportCtrl)	I	S
TU88	Viene verificato che il metodo onUpload non ritorni il modello aspettato. (ImportCtrl)	I	S
TU89	Viene verificato che il metodo onUpload ritorni un errore se il predittore caricato non dovesse essere valido. (ImportCtrl)	I	S
TU90	Viene verificato che il metodo loadDataSources carichi le sorgenti di dati correttamente e procede allo step 2. (ImportCtrl)	I	S
TU91	Viene verificato che il metodo loadDataSources non procede allo step 2 senza caricare sorgenti. (ImportCtrl)	I	S
TU92	Viene verificato che il metodo saveDashboardfunzioni correttamente per il salvataggio della dashboard. (ImportCtrl)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU93	Viene verificato che PredictCtrl venga costruito correttamente. (PredictCtrl)	I	S
TU94	Viene verificato che il metodo verifyDashboard ritorni una dashboard vuota. (PredictCtrl)	I	S
TU95	Viene verificato che il metodo verifyDashboard ritorni una dashboard corretta. (PredictCtrl)	I	S
TU96	Viene verificato che il metodo verifyDashboard ritorni false se non esistesse la dashboard. (PredictCtrl)	I	S
TU97	Viene verificato che il metodo resetButtonState imposti correttamente lo stato quando si è a conoscenza dello stato. (PredictCtrl)	I	S
TU98	Viene verificato che il metodo resetButtonState venga impostato a “stop” se lo stato non è stato definito. (PredictCtrl)	I	S
TU99	Viene verificato che il metodo stopPrediction imposti il corretto stato del Button se lo stato della predizione dovesse essere definito. (PredictCtrl)	I	S
TU100	Viene verificato che il metodo stopPrediction imposti il corretto stato del Button se lo stato della predizione non dovesse essere definito. (PredictCtrl)	I	S
TU101	Viene verificato che il metodo getPanelsState legga lo stato della predizione quando essa è spenta. (PredictCtrl)	I	S
TU102	Viene verificato che il metodo getPanelsState legga il refresh time della predizione quando impostato in secondi. (PredictCtrl)	I	S
TU103	Viene verificato che il metodo getPanelsState legga il refresh time della predizione quando non è impostata in secondi. (PredictCtrl)	I	S
TU104	Viene verificato che il metodo getPanelsState legga il refresh time della predizione quando non viene impostata. (PredictCtrl)	I	S
TU105	Viene verificato che il metodo timeToMilliseconds converta in millisecondi quando viene passato un parametro in secondi. (PredictCtrl)	I	S
TU106	Viene verificato che il metodo timeToMilliseconds converta in millisecondi quando viene passato un parametro in minuti. (PredictCtrl)	I	S
TU107	Viene verificato che il metodo timeToMilliseconds non converta nulla nel caso in cui non gli venga passato alcun dato. (PredictCtrl)	I	S
TU108	Viene verificato che il metodo startPrediction dia errore quando viene avviata la predizione su una dashboard vuota. (PredictCtrl)	I	S
TU109	Viene verificato che il metodo startPrediction dia errore quando viene avviata la predizione con un refresh time non valido. (PredictCtrl)	I	S
TU110	Viene verificato che il metodo startPrediction invii una notifica di conferma nel caso in cui venga avviata la predizione. (PredictCtrl)	I	S
TU111	Viene verificato che il metodo stopPrediction invii una notifica di conferma nel caso in cui venga stoppata la predizione. (PredictCtrl)	I	S
TU112	Viene verificato che il metodo redirect permetta di reindirizzare correttamente la dashboard. (PredictCtrl)	I	S
TU113	Viene verificato che il metodo redirect non reindirizzi nel caso in cui non esistesse la dashboard. (PredictCtrl)	I	S
TU114	Viene verificato che il metodo removePanel rimuova correttamente il pannello specificato dalla dashboard. (PredictCtrl)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU115	Viene verificato che Builder venga costruito correttamente. (Builder)	I	S
TU116	Viene verificato che il metodo buildTarget costruisca correttamente un Target. (Builder)	I	S
TU117	Viene verificato che il metodo buildView costruisca correttamente un View quando il modello è SVM. (Builder)	I	S
TU118	Viene verificato che il metodo buildView costruisca correttamente un View quando il modello è RL. (Builder)	I	S
TU119	Viene verificato che Dashboard venga costruito correttamente quando viene passata la configurazione del database. (Dashboard)	I	S
TU120	Viene verificato che Dashboard venga costruito correttamente quando non viene passata la configurazione del database. (Dashboard)	I	S
TU121	Viene verificato che il metodo setThreshold imposti correttamente la soglia quando il pannello è di tipo grafico. (Dashboard)	I	S
TU122	Viene verificato che il metodo setThreshold imposti correttamente la soglia quando il pannello non è di tipo grafico (single stat) . (Dashboard)	I	S
TU123	Viene verificato che il metodo setThreshold imposti correttamente la soglia quando il pannello è di tipo single state e l'operazione è di tipo greater than. (Dashboard)	I	S
TU124	Viene verificato che il metodo setThreshold imposti correttamente la soglia quando il pannello è di tipo single state e l'operazione non è di tipo greater than (less than, equal). (Dashboard)	I	S
TU125	Viene verificato che il metodo setAlert imposti correttamente l'allert quando il pannello è di tipo grafico. (Dashboard)	I	S
TU126	Viene verificato che il metodo setAlert imposti correttamente l'allert quando il pannello è di tipo single stat. (Dashboard)	I	S
TU127	Viene verificato che il metodo removeThreshold rimuova correttamente la soglia quando definita, se il pannello è di tipo single stat. (Dashboard)	I	S
TU128	Viene verificato che il metodo removeThreshold rimuova correttamente la soglia quando definita, se il pannello è di tipo grafico. (Dashboard)	I	S
TU129	Viene verificato che il metodo removeThreshold rimuova correttamente la soglia quando non è definita. (Dashboard)	I	S
TU130	Viene verificato che il metodo removeAll rimuova correttamente l'allert se definito. (Dashboard)	I	S
TU131	Viene verificato che il metodo removeAll non rimuova alcun allert nel caso in cui non ne esistesse alcuno. (Dashboard)	I	S
TU132	Viene verificato che il metodo addPanel che aggiunga correttamente il pannello. (Dashboard)	I	S
TU133	Viene verificato che il metodo storeSettings salvi correttamente la configurazione della dashboard. (Dashboard)	I	S
TU134	Viene verificato che il metodo updateSettings aggiorni correttamente la configurazione della dashboard nel caso in cui il numero di pannelli corrisponda al numero di variabili in Grafana. (Dashboard)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU135	Viene verificato che il metodo updateSettings aggiorni correttamente la configurazione della dashboard nel caso in cui il numero di pannelli non corrisponda al numero di variabili in Grafana, ma aggiornando le variabili di Grafana. (Dashboard)	I	S
TU136	Viene verificato che il metodo getJSON ritorni la dashboard in formato JSON. (Dashboard)	I	S
TU137	Viene verificato che il metodo removePanel rimuova correttamente il pannello. (Dashboard)	I	S
TU138	Viene verificato che il metodo setPredictionStarted imposti lo stato della predizione del pannello. (Dashboard)	I	S
TU139	Viene verificato che DBConnection venga costruito correttamente nel caso in cui vengano passati valori corretti. (DBConnection)	I	S
TU140	Viene verificato che DBConnection non venga costruito correttamente, inviando come errore “valori incorretti”, se gli venisse passato un host non corretto. (DBConnection)	I	S
TU141	Viene verificato che DBConnection non venga costruito correttamente, inviando come errore “valori incorretti”, se gli venisse passato una porta non corretta. (DBConnection)	I	S
TU142	Viene verificato che DBConnection non venga costruito correttamente, inviando come errore “valori incorretti”, se non gli venisse passato un database. (DBConnection)	I	S
TU143	Viene verificato che DBConnection non venga costruito correttamente, inviando come errore “valori incorretti” se non gli venissero passati valori. (DBConnection)	I	S
TU144	Viene verificato che GrafanaApiQuery venga costruito correttamente. (GrafanaApiQuery)	I	S
TU145	Viene verificato che il metodo getDataSources legga correttamente le sorgenti di dati disponibili. (GrafanaApiQuery)	I	S
TU146	Viene verificato che il metodo postDataSources aggiunga correttamente una sorgente di dati. (GrafanaApiQuery)	I	S
TU147	Viene verificato che il metodo getDashboard legga correttamente le dashboard disponibili. (GrafanaApiQuery)	I	S
TU148	Viene verificato che il metodo getFolder legga correttamente le cartelle disponibili. (GrafanaApiQuery)	I	S
TU149	Viene verificato che il metodo postDashboard aggiunga correttamente una dashboard	I	S
TU150	Viene verificato che il metodo getAllerts legga correttamente tutti gli alert impostati. (GrafanaApiQuery)	I	S
TU151	Viene verificato che il metodo postAlert aggiunga correttamente un alert. (GrafanaApiQuery)	I	S
TU152	Viene verificato che il metodo updateAlert aggiorni correttamente un alert impostato in precedenza. (GrafanaApiQuery)	I	S
TU153	Viene verificato che Influx venga creato correttamente. (Influx)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU154	Viene verificato che il metodo query ritorni un array contenente il risultato della “query” . (Influx)	I	S
TU155	Viene verificato che il metodo query ritorni un array vuoto nel caso in cui non vada a buon fine. (Influx)	I	S
TU156	Viene verificato che il metodo getLastValue ritorni un array contenente l’ultimo valore memorizzato nel database. (Influx)	I	S
TU157	Viene verificato che il metodo getLastValue ritorni un array vuoto nel caso in cui la chiamata non andasse a buon fine. (Influx)	I	S
TU158	Viene verificato che il metodo getLastValue ritorni un array contenente l’ultimo valore memorizzato nel database, ma l’istanza del database su cui è teoricamente connesso tramite istanza non è stata definita. (Influx)	I	S
TU159	Viene verificato che il metodo getLastValue ritorni un array vuoto, ma l’istanza del database su cui è teoricamente connesso tramite istanza non è stata definita. (Influx)	I	S
TU160	Viene verificato che il metodo getSources ritorni un array contenente le sorgenti disponibili. (Influx)	I	S
TU161	Viene verificato che il metodo getSources ritorni un array vuoto nel caso in cui le sorgenti non siano disponibili. (Influx)	I	S
TU162	Viene verificato che il metodo getSources non ritorni nulla nel caso in cui non dovesse trovare le sorgenti richieste. (Influx)	I	S
TU163	Viene verificato che il metodo getSorces invii un errore nel caso in cui la richiesta delle sorgenti non dovesse andare a buon fine. (Influx)	I	S
TU164	Viene verificato che il metodo getInstance ritorni un array contenente le istanze disponibili. (Influx)	I	S
TU165	Viene verificato che il metodo getInstance ritorni un errore nel caso in cui la richiesta delle istanze non dovesse andare a buon fine. (Influx)	I	S
TU166	Viene verificato che il metodo storeValue salvi sul database il valore fornito. (Influx)	I	S
TU167	Viene verificato che il metodo storeValue invii un errore nel caso in cui la scrittura non andasse a buon fine. (Influx)	I	S
TU168	Viene verificato che il metodo deletePrediction rimuova correttamente “measurement” (la tabella contenente le predizioni misurate) nel caso esista. (Influx)	I	S
TU169	Viene verificato che il metodo deletePrediction invii un errore nel caso la rimozione non dovesse andare a buon fine per situazioni anomale. (Influx)	I	S
TU170	Viene verificato che il metodo deletePrediction non rimuova ed invii un errore nel caso non esista “measurement” (la tabella contenente le predizioni misurate). (Influx)	I	S
TU171	Viene verificato che deleteAllPredictions rimuova tutte le predizioni. (Influx)	I	S
TU172	Viene verificato che Panel venga costruito correttamente. (Panel)	I	S
TU173	Viene verificato che il metodo getJSON ritorni correttamente la rappresentazione del pannello in formato JSON. (Panel)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU174	Viene verificato che PredictLooper venga costruito correttamente. (PredictLooper)	I	S
TU175	Viene verificato che il metodo setBackendSrv imposti correttamente la configurazione della predizione. (PredictLooper)	I	S
TU176	Viene verificato che il metodo setConfig imposti correttamente la configurazione. (PredictLooper)	I	S
TU177	Viene verificato che il metodo setConfig imposti correttamente la prima configurazione. (PredictLooper)	I	S
TU178	Viene verificato che il metodo setInflux imposti correttamente il riferimento al database. (PredictLooper)	I	S
TU179	Viene verificato che il metodo dbWrite scriva correttamente il risultato della predizione sul database. (PredictLooper)	I	S
TU180	Viene verificato che il metodo startPrediction avvii correttamente la predizione nel caso in cui essa sia stata definita. (PredictLooper)	I	S
TU181	Viene verificato che il metodo startPrediction non avvii la predizione nel caso in cui la predizione non sia stata definita. (PredictLooper)	I	S
TU182	Viene verificato che il metodo stopPrediction fermi la predizione. (PredictLooper)	I	S
TU183	Viene verificato che il metodo getPrediction ritorni la predizione che ci si aspetta nel caso il modello fosse SVM	I	S
TU184	Viene verificato che il metodo getPrediction ritorni la predizione che ci si aspetta nel caso il modello fosse RL. (PredictLooper)	I	S
TU185	Viene verificato che il metodo predictSVM chiami correttamente la funzione di predizione. (PredictLooper)	I	S
TU186	Viene verificato che il metodo predictRL chiami correttamente la funzione di predizione. (PredictLooper)	I	S
TU187	Viene verificato che predict ritorni la predizione che ci si aspetta. (RIAdapter)	I	S
TU188	Viene verificato che predictClass ritorni la predizione che ci si aspetta. (SvmAdapter)	I	S
TU189	Viene verificato che Target venga costruito correttamente. (Target)	I	S
TU190	Viene verificato che il metodo setId imposti correttamente l'id del target. (Target)	I	S
TU191	Viene verificato che il metodo getId legga correttamente l'id del target. (Target)	I	S
TU192	Viene verificato che il metodo getJSON ritorni correttamente il target in formato JSON nel caso in cui l'id sia stato settato. (Target)	I	S
TU193	Viene verificato che il metodo getJSON non ritorni nulla nel caso in cui l'id non sia stato settato correttamente. (Target)	I	S
TU194	Viene verificato che View venga costruito correttamente come tipo single stat. (Target)	I	S
TU195	Viene verificato che View venga costruito correttamente come tipo grafico. (Target)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU196	Viene verificato che il metodo setType imposti correttamente il tipo di pannello. (Target)	I	S
TU197	Viene verificato che il metodo setType non imposti il tipo di pannello nel caso il dato passatogli non fosse valido. (Target)	I	S
TU198	Viene verificato che il metodo setTitle imposti correttamente il titolo del pannello. (Target)	I	S
TU199	Viene verificato che il metodo setTitle non imposti il titolo nel caso in cui il dato passatogli non fosse valido. (Target)	I	S
TU200	Viene verificato che il metodo setId imposti correttamente l'id del pannello. (Target)	I	S
TU201	Viene verificato che il metodo setId non imposti l'id nel caso in cui il dato passatogli non fosse valido. (Target)	I	S
TU202	Viene verificato che il metodo setDataSource imposti correttamente la data source del pannello. (Target)	I	S
TU203	Viene verificato che il metodo setDataSource non imposti la data source nel caso in cui il dato passatogli non fosse valido. (Target)	I	S
TU204	Viene verificato che il metodo setDescription imposti correttamente la descrizione del pannello. (Target)	I	S
TU205	Viene verificato che il metodo setDescription non imposti la descrizione nel caso in cui il dato passatogli non fosse valido. (Target)	I	S
TU206	Viene verificato che il metodo setBackground imposti correttamente il background del pannello. (Target)	I	S
TU207	Viene verificato che il metodo getType ritorni correttamente il tipo del pannello. (Target)	I	S
TU208	Viene verificato che il metodo getTitle ritorni correttamente il titolo del pannello. (Target)	I	S
TU209	Viene verificato che il metodo getId ritorni correttamente l'id del pannello. (Target)	I	S
TU210	Viene verificato che il metodo getDataSource ritorni correttamente la data source del pannello. (Target)	I	S
TU211	Viene verificato che il metodo getDescription ritorni correttamente la descrizione del pannello. (Target)	I	S
TU212	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo grafico con la definizione del titolo. (Target)	I	S
TU213	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo grafico senza la definizione del titolo. (Target)	I	S
TU214	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo single stat con la definizione del titolo e la soglia definita. (Target)	I	S
TU215	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo single stat con la definizione del titolo e la soglia non definita. (Target)	I	S

TABELLA 6 CONTINUAZIONE

Codice	Descrizione	Stato	Esito
TU216	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo single stat senza la definizione del titolo e la soglia definita. (Target)	I	S
TU217	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo single stat senza la definizione del titolo e senza la soglia definita. (Target)	I	S
TU218	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo single stat senza la definizione della descrizione, ma con la definizione del titolo e della soglia definita. (Target)	I	S
TU219	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo single stat senza la definizione della descrizione, senza la definizione del titolo, ma con la soglia definita. (Target)	I	S
TU220	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo grafico senza la definizione del titolo e la soglia definita. (Target)	I	S
TU221	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo grafico senza la definizione del titolo e senza la soglia definita. (Target)	I	S
TU222	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo grafico senza la definizione della descrizione, ma con la definizione del titolo e della soglia definita. (Target)	I	S
TU223	Viene verificato che il metodo getJSON ritorni correttamente la view in formato JSON nel caso in cui il pannello sia di tipo grafico senza la definizione della descrizione, senza la definizione del titolo, ma con la soglia definita. (Target)	I	S

TABELLA 6 CONTINUAZIONE

2.2.8 Mock

Viene riportato l'elenco dei mock creati, al fine di riprodurre il comportamento di oggetti inaccessibili (in quanto non è possibile averne il controllo dell'implementazione), in modo controllato.

Codice	Descrizione	Stato	Esito
Addestramento			
M1	Creazione mock regression	I	S
M2	Creazione mock svm	I	S
M3	Creazione mock CsvReader	I	S
M4	Creazione mock RPreddittore	I	S
M5	Creazione mock WPreddittore	I	S
M6	Creazione mock RIAdapter	I	S
M7	Creazione mock SvmAdapter	I	S
M8	Creazione mock node_modules express	I	S
M9	Creazione mock node_modules formidable	I	S
M10	Creazione mock node_modules fs	I	S
M11	Creazione mock node_modules mime	I	S
M12	Creazione mock node_modules nconf	I	S
M13	Creazione mock node_modules path	I	S
Plug-in			
M14	Creazione mock di backendSrv	I	S
M15	Creazione mock di JQuery	I	S
M16	Creazione mock \$location	I	S
M17	Creazione mock events	I	S
M18	Creazione mock scope	I	S
M19	Creazione mock regression	I	S
M20	Creazione mock svm	I	S
M21	Creazione mock RIAdapter	I	S
M22	Creazione mock SvmAdapter	I	S
M23	Creazione mock Dashboard	I	S
M24	Creazione mock GrafanaQuery	I	S
M25	Creazione mock Influx	I	S
M26	Creazione mock RPreddittore	I	S
M27	Creazione mock Target	I	S
M28	Creazione mock View	I	S

TABELLA 7 - MOCK

3 Qualità di processo

I processi di sviluppo utilizzati durante la creazione di un prodotto software hanno una significativa influenza sulla qualità del prodotto stesso. Una buona gestione e il miglioramento della qualità dei processi può infatti portare più facilmente a un software di buona qualità. Per raggiungere questo obiettivo si è scelto di attenersi allo standard *ISO/IEC 15504*, chiamato anche SPICE, e di seguire lo schema del CICLO DI DEMINGG (o PDCAg) che fornisce un approccio per il controllo e il miglioramento continuo di processi e prodotti.

La descrizione generale di ciascuna metrica di Qualità di Processo è contenuta nella sezione delle *Norme di Progetto* che definisce il processo in questione, nella sezione seguente vengono specificati gli obiettivi per ciascuna metrica analizzata.

3.1 Obiettivi delle metriche

Categoria Processi	Processo	Nome metrica	Intervallo accettabile	Intervallo desiderabile
Processi Primari	Analisi dei requisiti	Percentuale requisiti obbligatori soddisfatti (PROS)	100%	100%
	Progettazione architetturale	Structural Fan-in (SFIN)	1	2
		Structural Fan-out (SFOUT)	[0,3]	0
	Progettazione di dettaglio	Coupling between objects (CBO)	[0, 7]	[0, 5]
Processi di supporto	Pianificazione	Schedule Variance (SV)	[-2, 2] ₁	[0, 2] ₂
		Budget Variance (BV)	[-10%, 10%] ₃	[0%, 10%] ₃
	Documentazione	Indice di Gulpease (IG)	[40, 100]	[60, 100]
		Correttezza ortografica	0	0
	Verifica	Copertura dei test-Function coverage (CT-FC)	[96%, 100%]	[98%, 100%]
		Copertura dei test-Statement coverage (CT-SC)	[95%, 100%]	[96%, 100%]
		Copertura dei test-Branch coverage (CT-BC)	[92%, 100%]	[95%, 100%]
		Copertura dei test-Condition coverage (CT-CC)	[90%, 100%]	[94%, 100%]

TABELLA 8 - OBIETTIVI DELLE METRICHE DELLA QUALITÀ DI PROCESSO

¹ Il numero di attività svolte è considerato accettabile se in confronto alle attività pianificate se ne sono svolte o 2 in meno o 2 in più. Si sono limitate le attività in anticipo per rispettare la qualità di ciò che si è prodotto.

² Il numero desiderabile di attività svolte in confronto alle attività pianificate può variare tra 0 e 2. Si sono limitate le attività in anticipo per rispettare la qualità di ciò che si è prodotto.

³ Margine di tolleranza dei costi sostenuti.

Categoria Processi	Processo	Nome metrica	Intervallo accettabile	Intervallo desiderabile
Processi di supporto	Verifica	Copertura dei test-Path coverage (CT-PC)	[96%, 100%]	[98%, 100%]
Processi organizzativi	Gestione qualità	Metriche soddisfatte (MS)	[70%, 100%]	[80%, 100%]
		Metriche pienamente soddisfatte (MPS)	[60%, 100%]	[70%, 100%]

TABELLA 8 CONTINUAZIONE

A Resoconto attività di verifica

In questa sezione dell'Appendice vengono posti gli esiti delle attività di verifica svolte sui documenti consegnati alle revisioni di avanzamento.

A.1 Analisi statica dei documenti

La tecnica di analisi WALKTHROUGHG permette l'individuazione di errori: questi vengono inseriti in una check list, al fine di ottenere una correzione più rapida e dunque una verifica più breve. Tramite questa lista sarà possibile l'uso della tecnica INSPECTIONG per le attività di verifica future.

A.2 Esiti verifiche della qualità di Prodotto

A.2.1 Funzionalità

- **Funzionalità sviluppate (FS) = 73%;**
 - NRS = 59;
 - NRO = 61;
 - NRF = 20;
 - **Esito: SUPERATO**
- **Correttezza funzionale (CF) = 85%**
 - NRC = 28
 - NPE = 33
 - **Esito: SUPERATO**

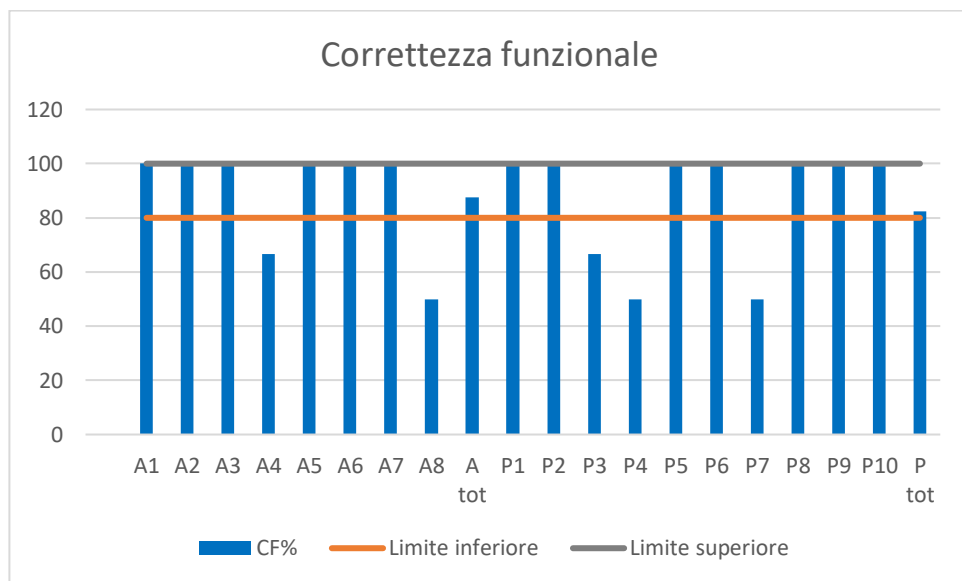


GRAFICO 1 - ESITI BUDGET CORRETTEZZA FUNZIONALE (A - ADDESTRAMENTO, P – PLUG-IN)

A.2.2 Affidabilità

- **Densità degli errori (DE) = 0%**
 - $N_{TF} = 0$
 - $N_{TE} = 294$
 - **Esito: SUPERATO**

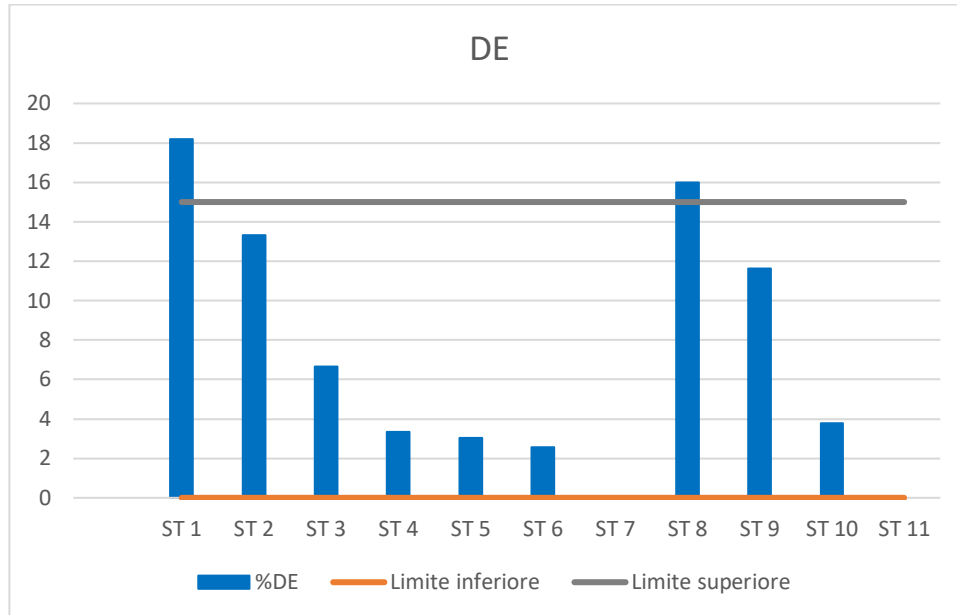


GRAFICO 2 - ESITI SESSIONI TEST (ST) IN RELAZIONE ALLA DENSITÀ DEGLI ERRORI

A.2.3 Efficienza

- **Tempo medio di risposta (TMR)**
 - **Parte addestramento:** 15 ms
 - **Parte plugin import:** 0.74 ms
 - **Esito: SUPERATO**
- **Consumo medio di potenza elaborativa (CMPE)**
 - **Parte addestramento:** 9-11%
 - **Parte plugin import:** 5%
 - **Esito: SUPERATO**
- **Consumo medio di memoria (CMM)**
 - **Parte addestramento:** 65 MB
 - **Parte plugin import:** 86.53 MB
 - **Esito: SUPERATO**

A.2.4 Usabilità

- **Ambiguità delle funzioni (AF) = 1%**
 - $N_{FA} = 1$
 - $N_F = 123$
 - **Esito: SUPERATO**
- **Ambiguità degli errori (AE) = 0%**
 - $N_{MEA} = 0$
 - $N_{ME} = 13$
 - **Esito: SUPERATO**

- **Descrizione delle funzioni (DF) = 100%**
 - $N_{FD} = 110$
 - $N_F = 110$
 - **Esito: SUPERATO**
- **Tempo medio di apprendimento (TMA) = 2h**
 - **Esito: SUPERATO**
- **Estetica dell'interfaccia (EI) = 9**
 - **Esito: SUPERATO**

A.2.5 Manutenibilità

- **Presenza di commenti (PC) = 27%**
 - $N_{RCC} = 704$
 - $N_{RC} = 2648$
 - **Esito: SUPERATO**
- **Complessità ciclomatica (CC) = 3 (valore medio di tutte le funzioni)**
 - **Esito: SUPERATO**
- **Impatto negativo delle modifiche (INM) = 9%**
 - $N_{ME} = 16$
 - $N_M = 171$
 - **Esito: SUPERATO**

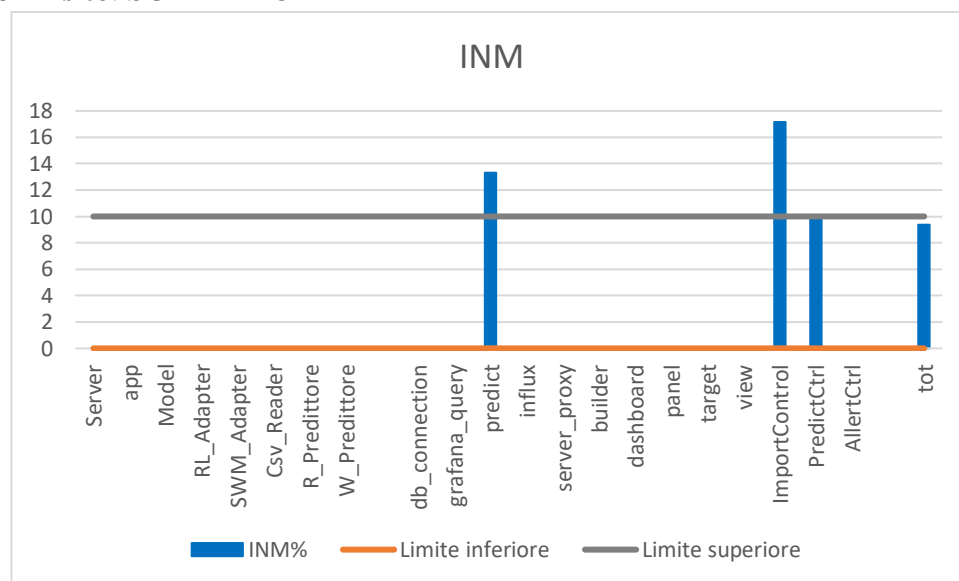


GRAFICO 3 - ESITI IMPATTO NEGATIVO MODIFICHE

- **Completamento dei test (CT) = 100%**
 - $N_{TE} = 325$
 - $N_{TCC} = 325$
 - **Esito: SUPERATO**

A.2.6 Portabilità

- **Browser supportati (BS)**
 - **Lista:** Chrome.32; Firefox.27, Safari 13.05, Opera 67.0.3575.115;
 - **Esito:** SUPERATO

A.3 Esiti verifiche della qualità di Processo

A.3.1 Analisi dei Requisiti

- **Percentuale requisiti obbligatori soddisfatti (PROS) =100%**
 - **NROS=** 44
 - **NROT=** 44
 - **Esito:** SUPERATO

A.3.2 Progettazione Architeturale

- **Structural Fan-in (SFIN) = 1**
 - **Esito:** SUPERATO

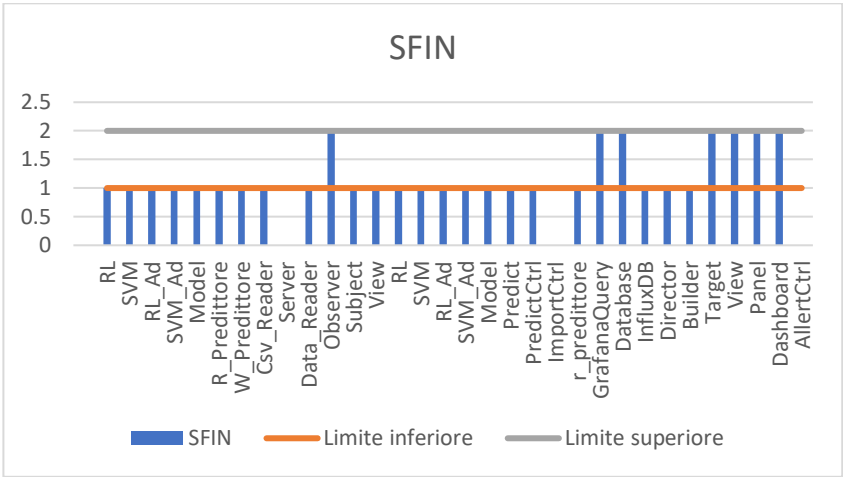


GRAFICO 4 - ESITI STRUCTURAL FAN-IN

- **Structural Fan-out (SFOUT) =1**
 - **Esito:** SUPERATO

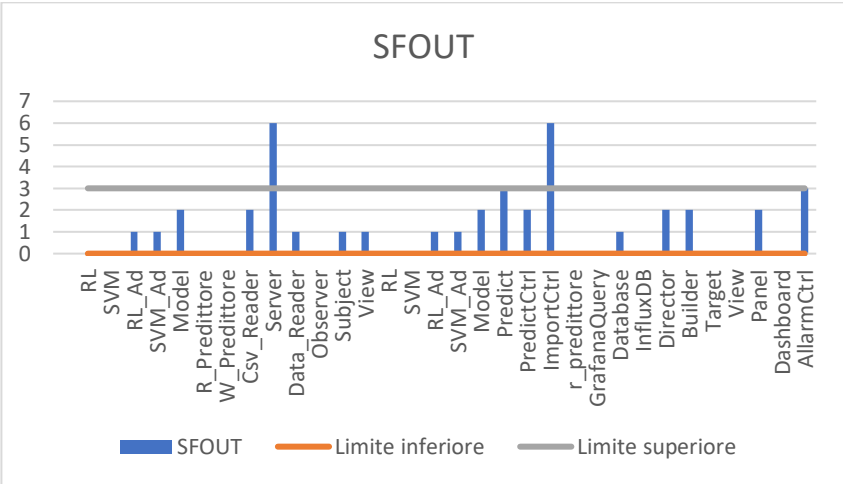


GRAFICO 5 - ESITI STRUCTURAL FAN-OUT

A.3.3 Progettazione di dettaglio

- **Coupling between objects (CBO) =3**
 - **Esito: SUPERATO**

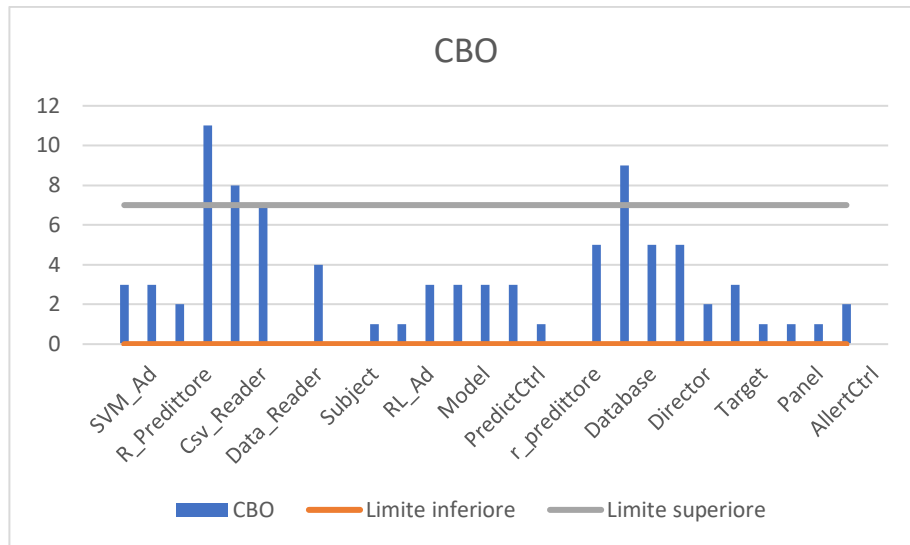


GRAFICO 6 - ESITI COUPLING BETWEEN OBJECTS

A.3.4 Pianificazione – Schedule Variance (SV)

- **Budget Cost of Work Performed (BCWP) =32**
- **Budget Cost of Work Scheduled (BCWS) =32**
- **Schedule Variance (SV) = 0**
 - **Esito: SUPERATO**

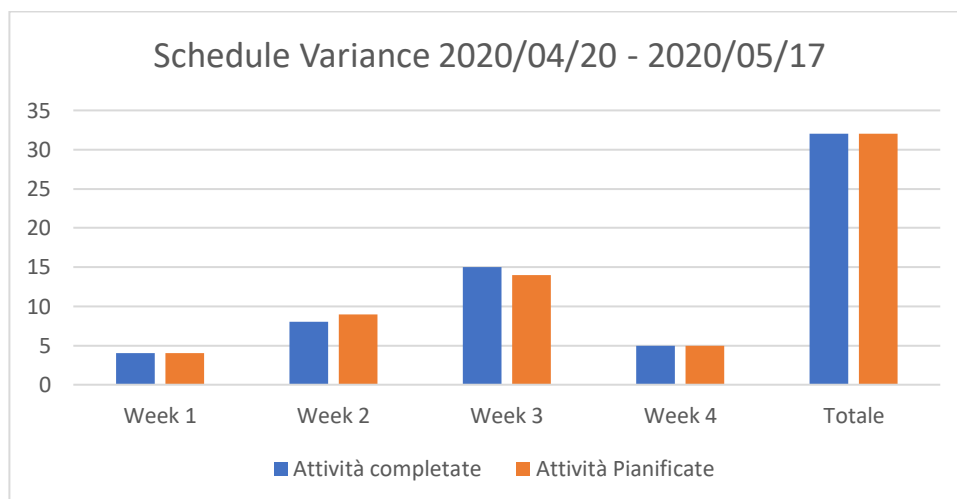


GRAFICO 7 - ESITI SCHEDULE VARIANCE

A.3.5 Pianificazione – Budget Variance (BV)

A.3.5.1 Periodo di Analisi

- **Budget Cost of Work Scheduled (BCWS):** 4.130 €
- **Actual Cost of Work performed (ACWP):** 4.365 €
- **BV:** -5,69%

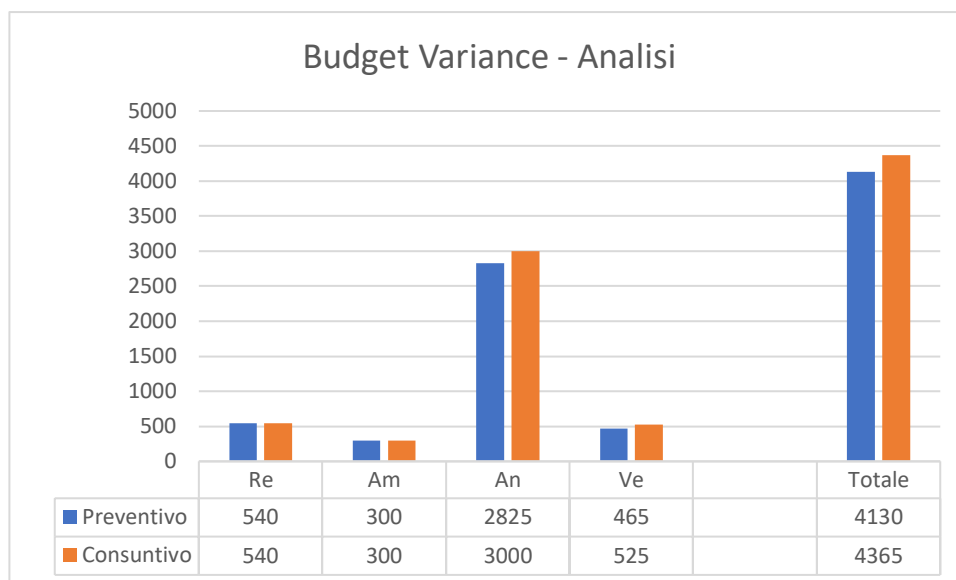


GRAFICO 8 - ESITI BUDGET VARIANCE PER IL PERIODO DI ANALISI

A.3.5.2 Periodo di Progettazione della Base Tecnologica

- **Budget Cost of Work Scheduled (BCWS):** 4.265 €
- **Actual Cost of Work performed (ACWP):** 4.094 €
- **BV:** +4,01%

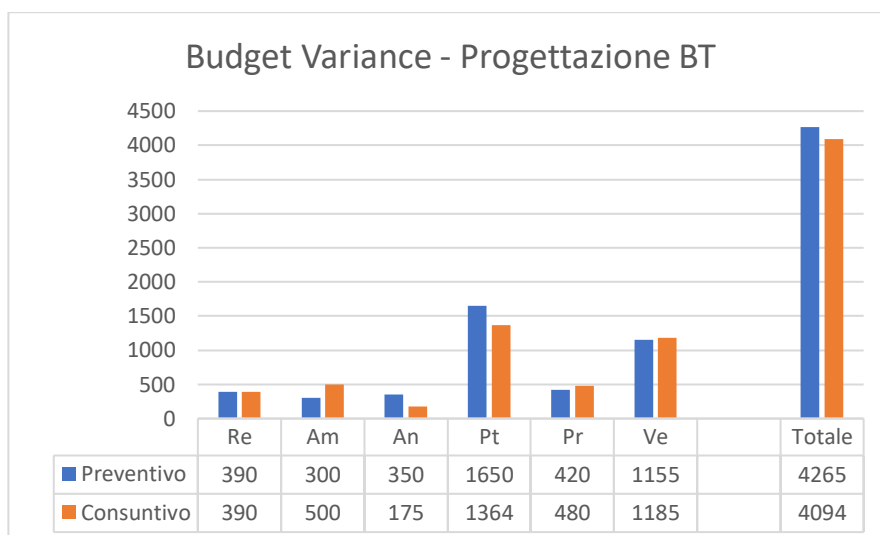
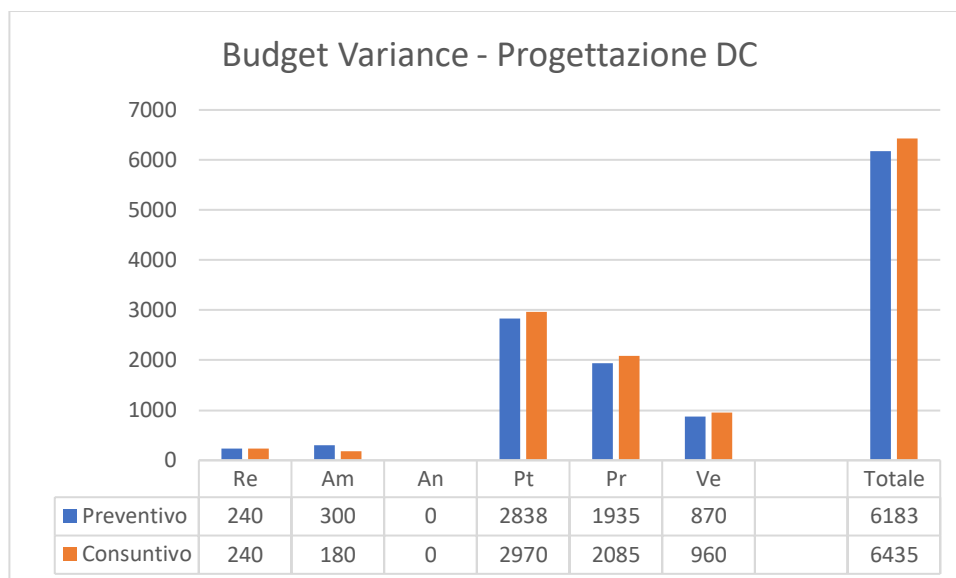


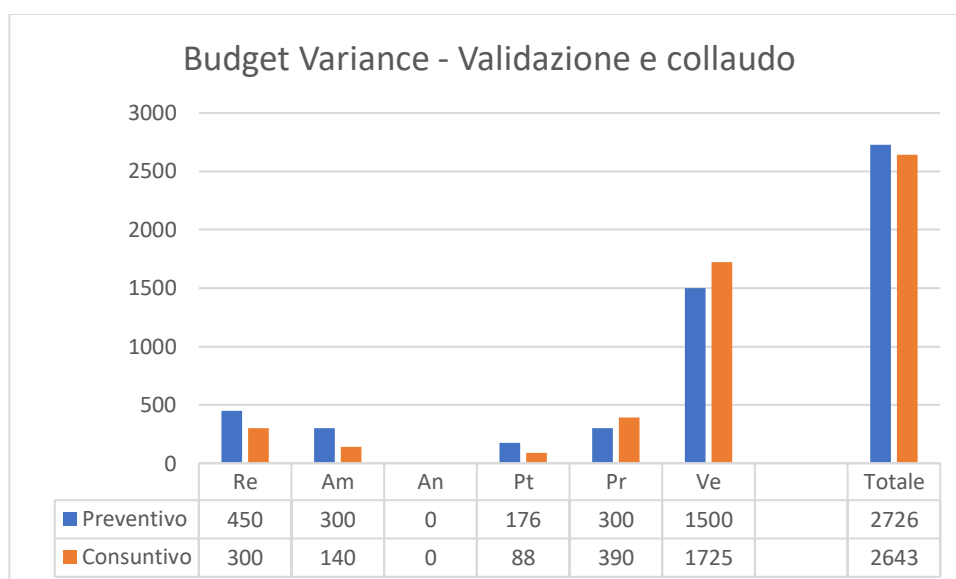
GRAFICO 9 - ESITI BUDGET VARIANCE PER LA PROGETTAZIONE BASE TECNOLOGICA

A.3.5.3 Periodo di Progettazione di dettaglio e codifica

- **Budget Cost of Work Scheduled (BCWS): 6.183 €**
- **Actual Cost of Work performed (ACWP): 6.435 €**
- **BV: -4,08%**

**GRAFICO 10 - ESITI BUDGET VARIANCE PER LA PROGETTAZIONE DI DETTAGLIO E CODIFICA****A.3.5.4 Periodo di Validazione e collaudo**

- **Budget Cost of Work Scheduled (BCWS): 2.643 €**
- **Actual Cost of Work performed (ACWP): 2.726 €**
- **BV: 3.04%**
 - **Esito: SUPERATO**

**GRAFICO 11 - ESITI BUDGET VARIANCE PER LA VALIDAZIONE E COLLAUDO**

A.3.5.5 Budget Variance Totale

- **Budget Cost of Work Scheduled (BCWS):** 17.304 €
- **Actual Cost of Work performed (ACWP):** 17.537 €
- **BV:** -1.35%
 - **Esito:** SUPERATO

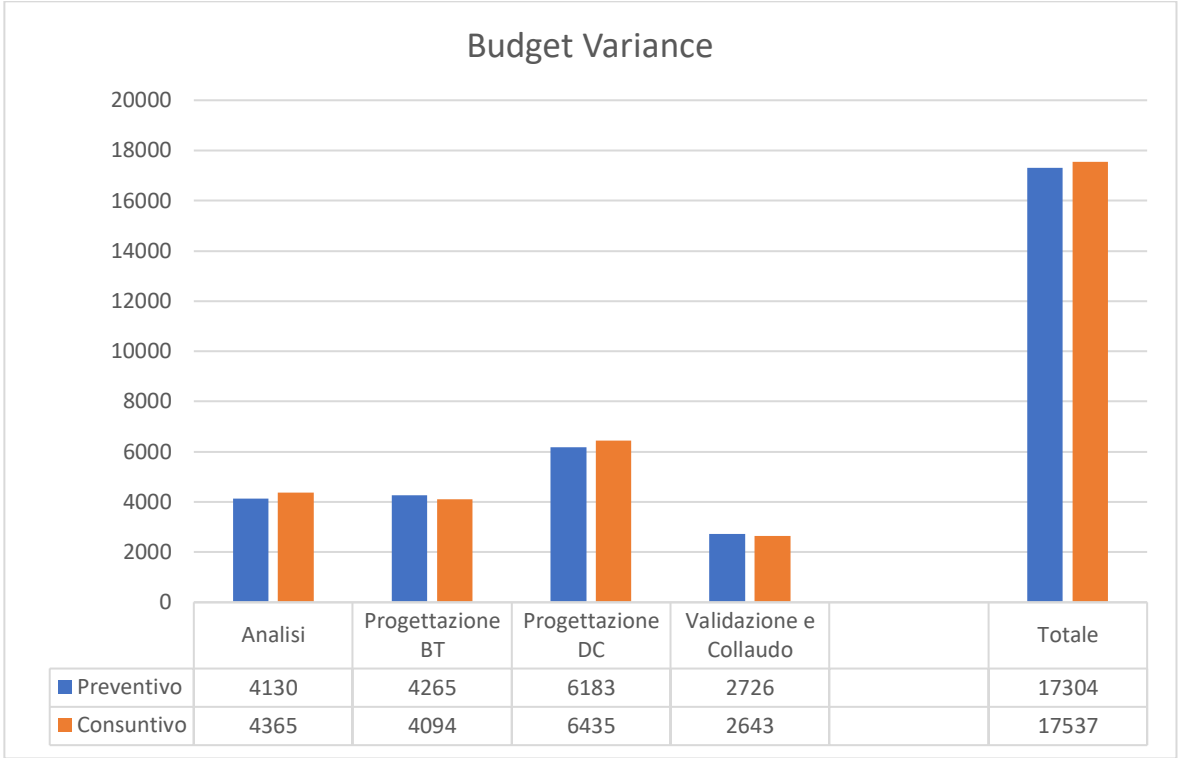


GRAFICO 12 - ESITI BUDGET VARIANCE

A.3.6 Documentazione

A.3.6.1 Indice di Gulpease (IG)

La Tabella 9 riporta gli INDICI DI GULPEASE dei documenti realizzati.

DOCUMENTO	DATA INIZIO	DATA FINE	INDICE GULPEASE	STATO
<i>NORME DI PROGETTO V.1.4.0</i>	2019/12/04	2020/05/11	56	Superato
<i>ANALISI DEI REQUISITI V.1.4.0</i>	2019/12/19	2020/05/11	67	Superato
<i>STUDIO DI FATTIBILITÀ V.1.4.0</i>	2019/12/10	2020/04/11	55	Superato
<i>PIANO DI PROGETTO V.1.4.0</i>	2019/12/05	2020/05/11	54	Superato
<i>PIANO DI QUALIFICA V.1.4.0</i>	2019/12/22	2020/05/11	59	Superato
<i>GLOSSARIO V.1.4.0</i>	2019/12/05	2020/05/11	61	Superato
<i>LETTERA DI PRESENTAZIONE</i>	2020/04/05	2020/05/11	94	Superato
<i>MANUALE UTENTE V.1.4.0</i>	2020/03/24	2020/05/11	60	Superato
<i>MANUALE SVILUPPATORE V.1.4.0</i>	2020/03/24	2020/05/11	58	Superato
<i>ALLEGATO TECNICO</i>	2020/04/03	2020/05/11	60	Superato
<i>VI12_2020-04-23</i>	2020/04/23	2020/05/11	51	Superato
<i>VI13_2020-05-06</i>	2020/05/06	2020/05/11	75	Superato
<i>VE11_2020-05-04</i>	2020/05/04	2020/05/11	80	Superato

TABELLA 9 - ESITI INDICI DI GULPEASE

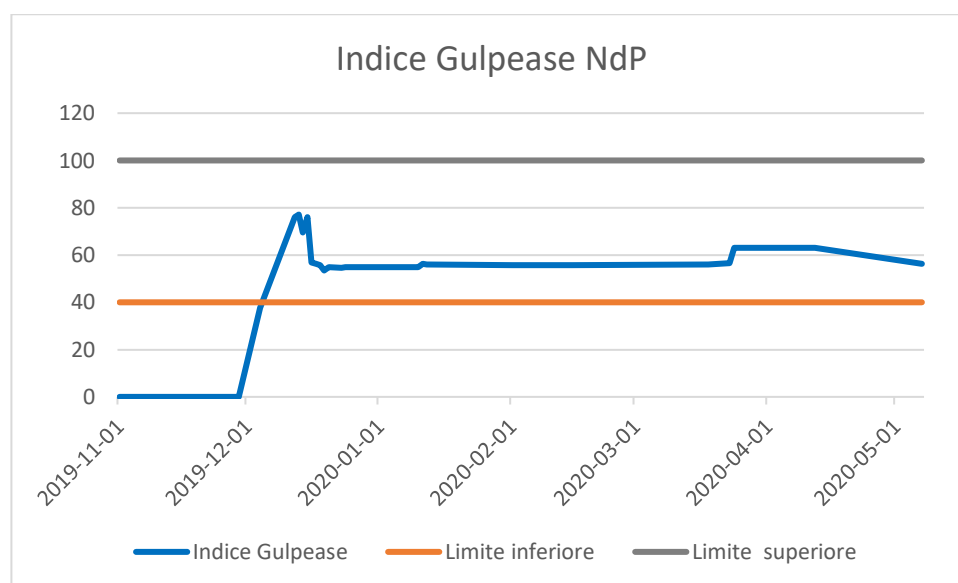


GRAFICO 13 – GRAFICO INDICE GULPEASE DELLE NORME DI PROGETTO

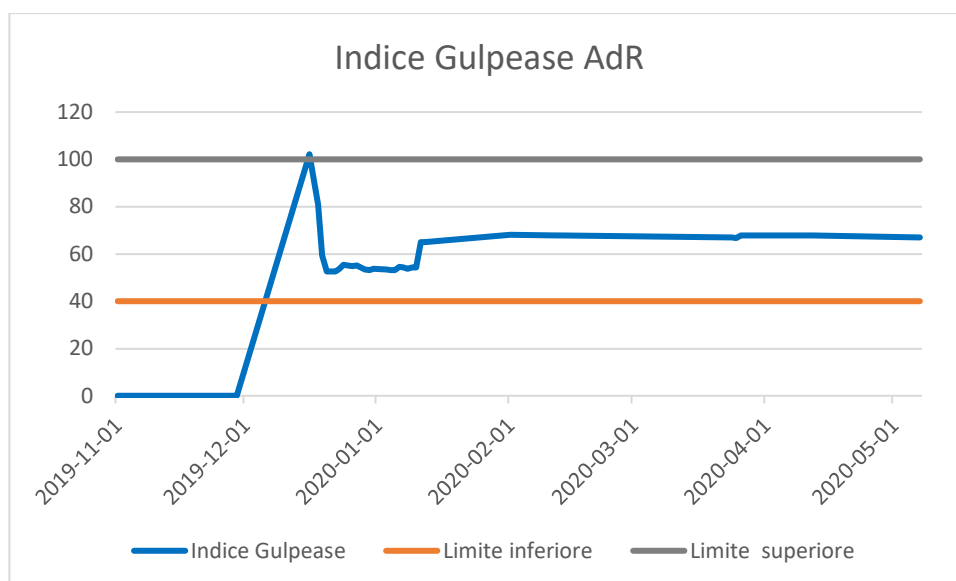


GRAFICO 14 – GRAFICO INDICE GULPEASE DELL'ANALISI DEI REQUISITI

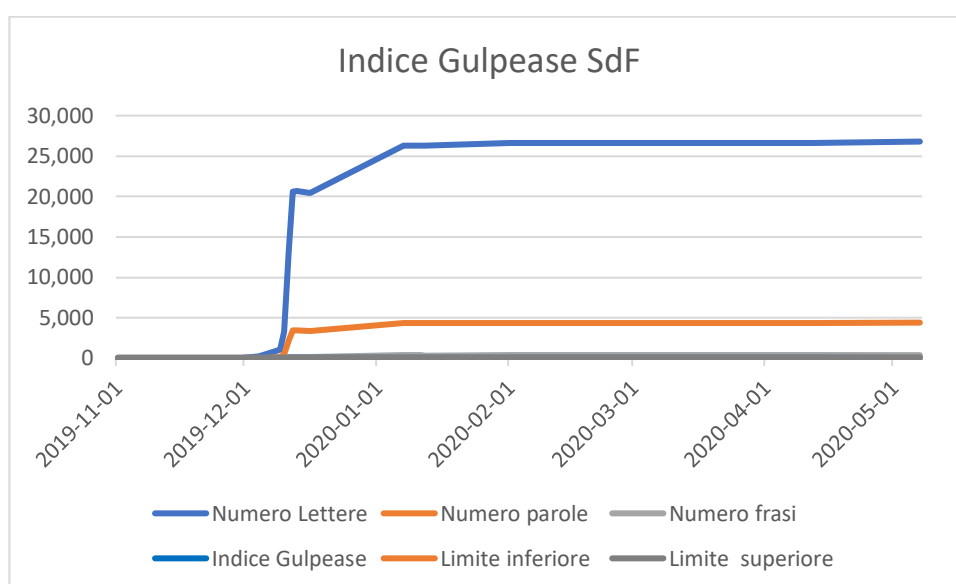


GRAFICO 15 – GRAFICO INDICE GULPEASE DELLO STUDIO DI FATTIBILITÀ

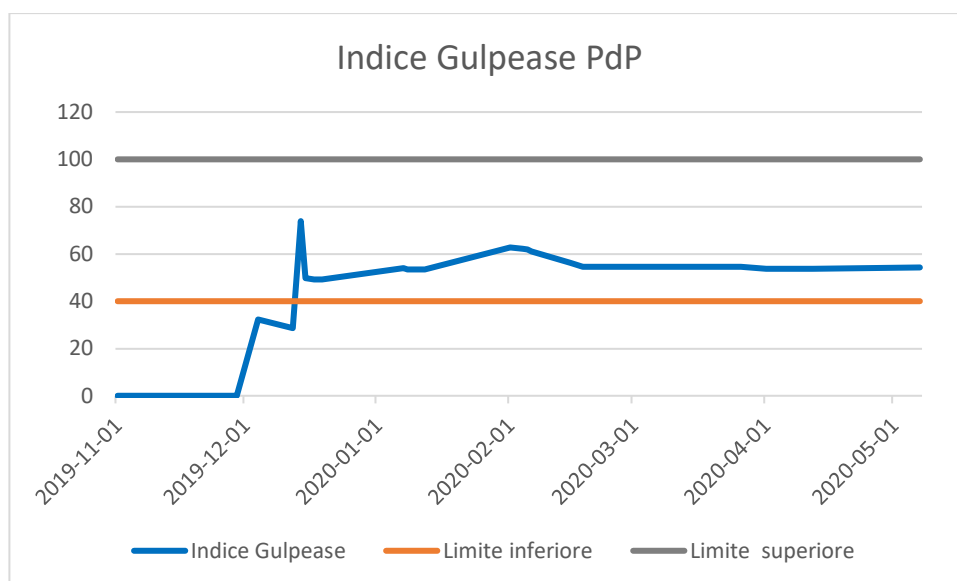


GRAFICO 16 – GRAFICO INDICE GULPEASE DEL PIANO DI PROGETTO

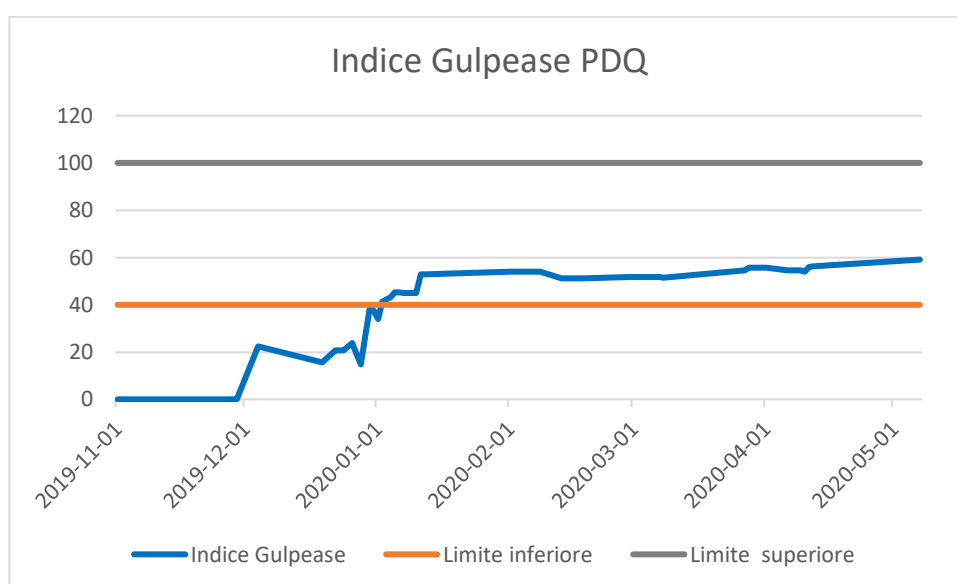


GRAFICO 17 – GRAFICO INDICE GULPEASE DEL PIANO DI QUALIFICA

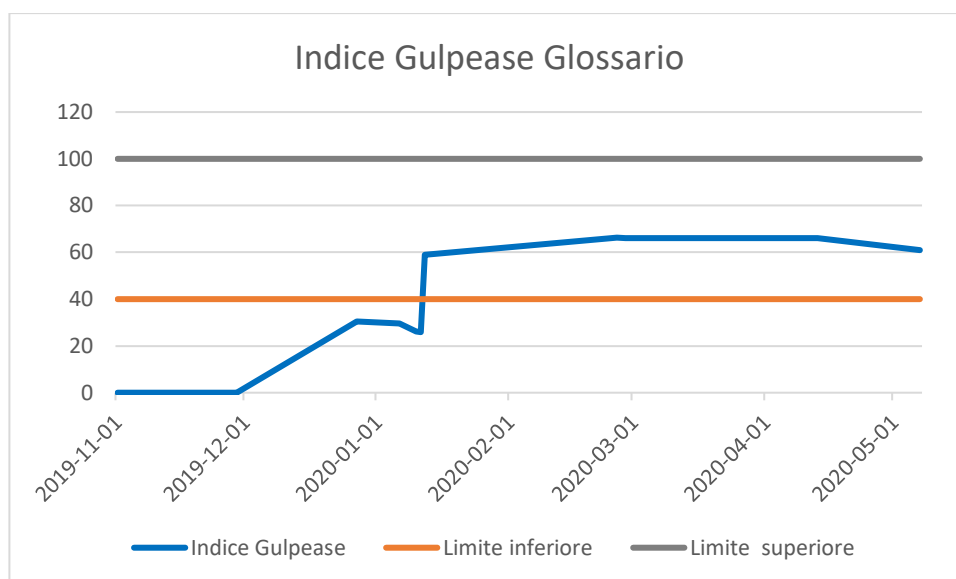


GRAFICO 19 – GRAFICO INDICE GULPEASE DEL GLOSSARIO

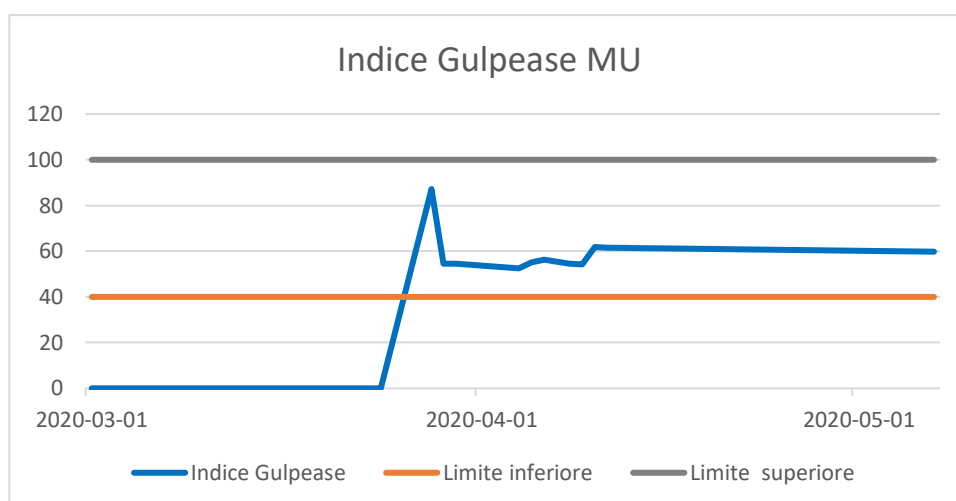


GRAFICO 20 – GRAFICO INDICE GULPEASE DEL MANUALE UTENTE

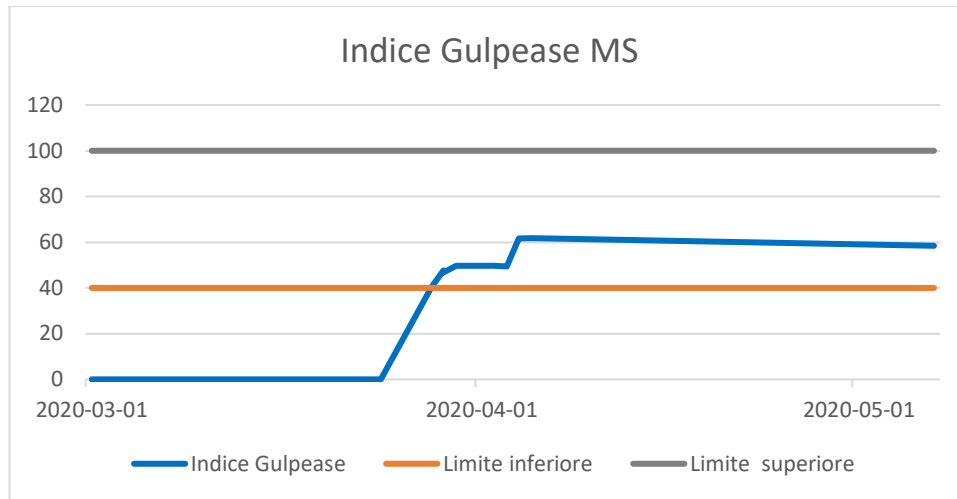


GRAFICO 21 – GRAFICO INDICE GULPEASE DEL MANUALE SVILUPPATORE

A.3.6.1 Correttezza Ortografica (IG)

- **Correttezza ortografica = 0**
 - **Esito: SUPERATO**

A.3.7 Verifica

A.3.7 Copertura dei Test

- **Function coverage=98%**
 - **Esito: SUPERATO**
- **Statement coverage =98%**
 - **Esito: SUPERATO**
- **Branch coverage =96%**
 - **Esito: SUPERATO**
- **Condition coverage =96%**
 - **Esito: SUPERATO**
- **Path coverage =97%**
 - **Esito: SUPERATO**

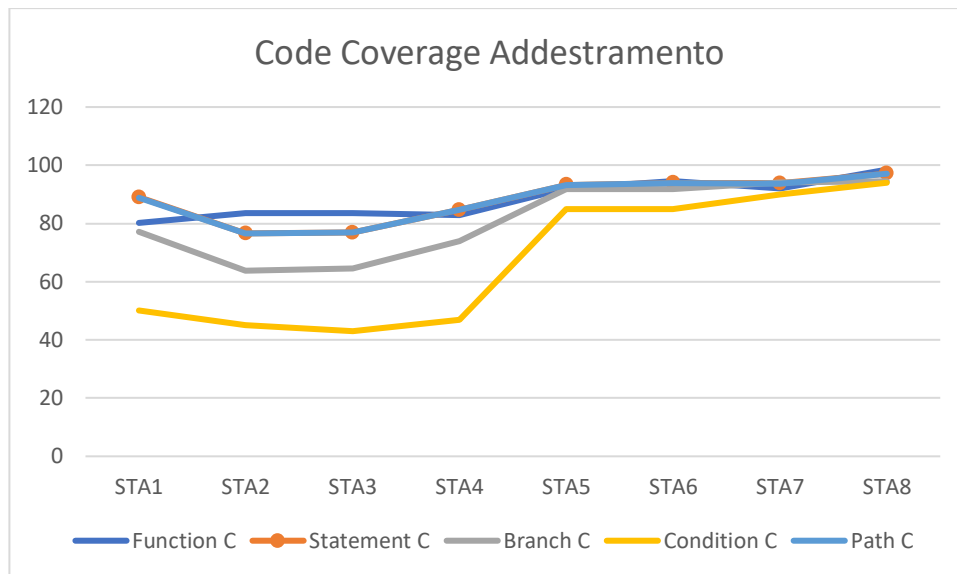


GRAFICO 22 – GRAFICO CODE COVERAGE ADDESTRAMENTO

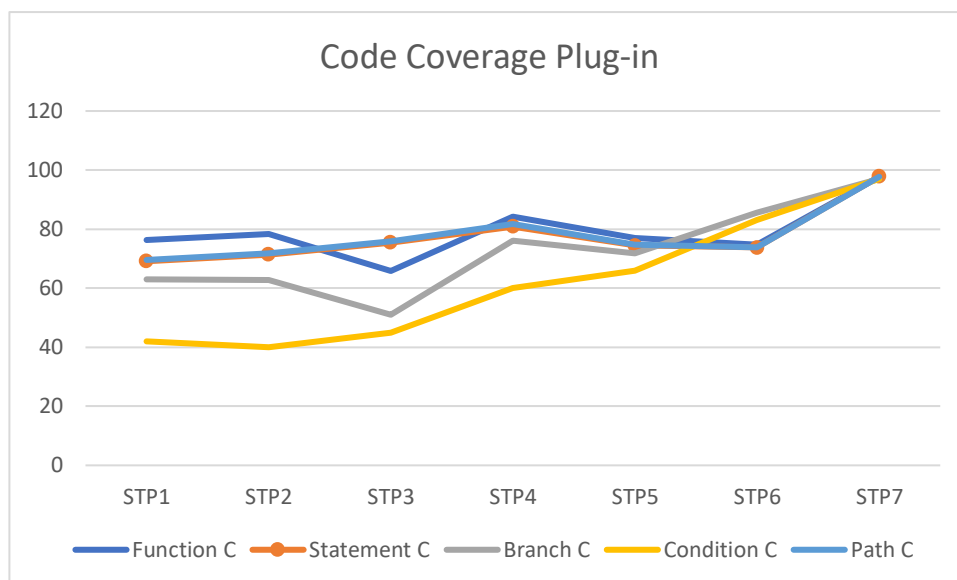


GRAFICO 23 – GRAFICO CODE COVERAGE PLUG-IN

A.3.7 Gestione Qualità

- **Metriche soddisfatte (MS) = 100%**
 - $N_{MS} = 29$
 - $N_M = 29$
 - **Esito: SUPERATO**
- **Metriche pienamente soddisfatte (MPS) = 83%**
 - $N_{MPS} = 24$
 - $N_M = 29$
 - **Esito: SUPERATO**

B Valutazioni per il miglioramento

In questa sezione dell'Appendice si riporta l'autovalutazione del gruppo Carbon12 in merito al processo produttivo svolto nei periodi di progetto. Al termine di ogni periodo viene svolta una discussione in merito ai problemi riscontrati e vengono valutate le modalità di risoluzione in modo efficiente onde evitare la ripetizione dei medesimi errori e che tali problematiche si ripresentino in futuro. L'esito dell'autovalutazione è riportato nei paragrafi seguenti.

I membri del gruppo hanno esternato le problematiche riscontrate durante il lavoro svolto e il confronto collegiale ha estrapolato tre ambiti principali:

- **Organizzazione:** problemi inerenti all'organizzazione del gruppo;
- **Ruoli:** problemi inerenti allo svolgimento di un ruolo;
- **Strumenti:** problemi inerenti all'uso degli strumenti impiegati.

Essendo questa un'autovalutazione non potrà essere pienamente accurata, come lo sarebbe un'analisi compiuta da un esperto esterno e nemmeno obiettiva o sufficientemente EFFICACE⁴; nonostante ciò porterà sicuramente dei contributi positivi e costruttivi nei processi lavorativi rendendo il team più SISTEMATICO⁴, DISCIPLINATO⁴, collaborativo, QUANTIFICANDO⁴ il lavoro svolto.

Per far sì che questo sistema funzioni, i membri del gruppo necessariamente devono essere onesti, esponendo i propri problemi e pensieri riguardanti le attività e metodologie lavorative applicate, al fine di solidificare la coesione tra tutti i componenti del gruppo.

Questa sezione verrà aggiornata se si dovessero riscontrare altri parametri di miglioramento necessari alla maturazione del team.

B.1 Fase di Analisi

B.1.1 Valutazione sull'organizzazione

Il problema riscontrato riguarda l'organizzazione degli incontri con tutti i membri del gruppo. Essendo il team composto da studenti e non da lavoratori non tutti sono sempre disponibili in caso fosse necessaria una riunione, sia per motivi di studio, per seguire un corso, per lavoro o anche per motivazioni personali.

Questa difficoltà si è presentata nel periodo iniziale dello svolgimento del lavoro, ma il gruppo si è adoperato in modo tale che gli incontri di persona fossero limitati alle occasioni in cui era strettamente necessario, utilizzando strumenti software di organizzazione e comunicazione, come Windows Teams. Tale software ha permesso di comunicare lo svolgimento delle attività, la pianificazione tramite un Planner e la possibilità di effettuare chiamate nelle quali era possibile anche la condivisione dello schermo.

Si è poi deciso che, nel qual caso il team ritenesse necessario un incontro con tutti i membri, la pianificazione di tale incontro deve essere antecedente di almeno due settimane, per permettere ai membri del team di organizzarsi in modo tale da essere liberi nel giorno stabilito.

Per quanto concerne gli incontri con membri esterni al gruppo, si è deciso in primis di valutare l'importanza degli argomenti da trattare e a seconda di ciò misurare un lasso di tempo sufficiente per il gruppo di elaborare tali problemi e per l'individuo esterno per organizzare una data per l'incontro.

⁴ Vedi *Quantificabile* nel Glossario Esterno

Il team ha poi stabilito che al momento dell'organizzazione dell'incontro, gli argomenti che verranno trattati (argomenti all'Ordine del Giorno, esposti nei verbali) vengono decisi in anticipo facendo uso di sondaggi, che permettano di classificare quali siano gli argomenti di maggior rilevanza assegnando delle priorità. Inoltre, questi sondaggi possono aiutare a definire anche la durata possibile dell'incontro, che sia virtuale o fisico.

B.1.2 Valutazione sui ruoli

Attualmente i membri del team hanno svolto le attività nei panni di Amministratore, Responsabile, Analista e Verificatore.

Vengono successivamente elencati difetti e problematiche riscontrate nella prima FASEG di analisi.

B.1.2.1 Amministratore

Il difetto riscontrato in questa figura sono i compiti che possiede, in quanto spesso è marginale, poiché il Responsabile quasi istintivamente assumeva i compiti dell'Amministratore. L'Amministratore è stato sicuramente utile nelle fasi iniziali per la scelta degli strumenti, ma il suo ruolo è stato quasi sovrastato dal Responsabile di progetto. Se necessario per motivi di budget la figura dell'Amministratore in alcune fasi del progetto potrebbe non comparire ed essere sostituito dal Responsabile.

B.1.2.2 Responsabile

La problematica emersa per il Responsabile riguarda il sovraccaricamento del lavoro assunto, come accennato nel paragrafo precedente il Responsabile ha inglobato il ruolo dell'Amministratore in questo periodo di analisi iniziale. A causa della carenza di esperienza, si è deciso di limitare il ruolo del Responsabile, suddividendo più accuratamente i compiti che gli concernono e in caso di farsi assistere da un Amministratore.

Il Responsabile si è spesso trovato in difficoltà nella gestione delle diverse opinioni del team, ciò però ha permesso una maturazione da parte di tutti i membri, in quanto i soggetti in contrasto si sono adoperati per gestire il problema collaborativamente, supervisionati, ove necessario, dal Responsabile per gestire la situazione in modo costruttivo senza far sentire a disagio nessuna delle due parti.

Infine, il Responsabile ha avuto difficoltà, dovute all'inesperienza, nella gestione del lavoro altrui, in quanto doveva assegnare le attività rispettando gli impegni che potevano avere i suoi collaboratori. Ciò ha portato una maturazione nei responsabili nella gestione collaborativa di un gruppo numeroso di persone.

B.1.2.3 Analista

Si sono riscontrate difficoltà nel decomporre e classificare i requisiti per lo studio svolto nell'Analisi dei Requisiti. Questa complicazione è dovuta alle poche conoscenze sull'argomento e sulla difficoltà nella sua analisi. Per ovviare a ciò tutto il team ha collaborato per ottenere la qualità di analisi desiderata.

B.1.2.4 Verificatore

Per i verificatori è sorta una complicazione nell'analisi dei documenti per la verifica della correttezza e completezza, per motivi di poca conoscenza degli argomenti trattati, ma anche nello stabilire le correzioni da apportare.

I Verificatori hanno perciò dedicato molta attenzione e studio degli argomenti trattati, ottenendo così un'analisi più veritiera.

B.1.3 Valutazione sull'uso degli strumenti

Le problematiche riguardanti gli strumenti usati sono di tipologia generale.

Un problema emerso riguarda il prestare attenzione alla numerazione del versionamento dei documenti Word. In uno degli ultimi incontri fatti dal team si è discusso di questa problematica stabilendo che i membri che fanno modifiche a tali documenti devono porre molta più attenzione ad aggiornare il numero di versione.

I componenti del team hanno poi riscontrato una certa difficoltà nella gestione dei commenti in Word e Drive, principalmente dovuti all'inesperienza. Un esempio di ciò che si è riscontrato su Drive è l'invio di e-mail a tutti i membri del team, anche se rivolti ad un unico individuo. Per quanto riguarda Windows Teams, i commenti non vengono notificati al soggetto interessato, per ovviare a ciò si è deciso di commentare senza specificare il nome e notificare al membro interessato tramite la chat dei canali appositamente creati a tal scopo.

Alcune volte le comunicazioni su *Microsoft Teams* venivano perse e dunque qualche membro rimaneva poco aggiornato: si è pertanto ricordato a tutti i membri di verificare frequentemente le novità presenti sulla piattaforma, in quanto per comunicazioni generali non viene menzionata una specifica persona ma viene scritto un messaggio normale.

B.2 Fase di Progettazione della base tecnologica

B.2.1 Valutazione sull'organizzazione

Il principale problema organizzativo riscontrato in questa fase è legato alle difficoltà del gruppo di organizzare riunioni in presenza. Inizialmente tale problema è sorto perché la fine delle lezioni ha fatto sì che molti membri del gruppo non avessero motivo di recarsi in sede ed inoltre il periodo di esami subito seguente ha comportato difficoltà nella conciliazione delle disponibilità dei membri del gruppo. Nelle ultime settimane inoltre la chiusura cautelativa straordinaria delle sedi universitarie legata all'emergenza del Covid-19 ha impedito al gruppo di usufruire degli spazi dell'ateneo per i propri incontri. Conseguentemente il gruppo si è adoperato per organizzare gli incontri in via telematica. Tale modalità ha comportato alcune difficoltà nella coordinazione e partecipazione attiva di tutti i componenti del team. Si è deciso quindi, anche per facilitare il compimento delle attività, di organizzare gruppi di lavoro opportunamente di 2, 3 o 4 componenti al massimo. In questo modo è stato più semplice anche organizzare riunioni telematiche a cui potessero partecipare tutti i componenti dei sotto-gruppi, perché vi era un minor numero di persone da coordinare. Regolarmente venivano poi organizzate delle call nei giorni e negli orari in cui vi era la disponibilità di tutti i componenti in modo che tutti venissero aggiornati sull'evoluzione delle attività.

Date le circostanze è risultato ancora più importante definire una pianificazione puntuale in cui venissero delineate le attività assegnate a ogni gruppo di lavoro oppure a una singola persona

mediante un uso efficiente del Planner che permettesse a tutti di conoscere lo stato di avanzamento del progetto. Inoltre, è stato fatto uso del sistema di ticketing di GitHub per quanto riguarda le attività legate allo sviluppo del PoC. Tuttavia, non sempre i componenti erano puntuali nell'aggiornare lo stato di progresso delle proprie attività e più volte è stato necessario sollecitare il gruppo nell'utilizzo corretto del Planner e del sistema di ticketing.

In generale, la scelta di decomporre il gruppo in sotto-gruppi di lavoro responsabili di diverse aree di sviluppo è risultata vincente, pertanto si è deciso di proseguire con questa modalità anche in seguito per la prosecuzione dello sviluppo del prodotto.

Per quanto riguarda gli incontri con partecipanti esterni, si è valutato che l'organizzazione regolare di incontri anche telematici con il proponente abbia fatto sì che lo sviluppo del PoC sia risultato conforme alle aspettative grazie ai continui feedback ricevuti. In questo modo le attività di correzione sono state molto limitate e sono stati raggiunti più obiettivi di quanti non ne fossero stati stimati inizialmente.

B.2.2 Valutazione sui ruoli

B.2.2.1 Amministratore

Rispetto alla fase precedente si è notato un netto miglioramento nello svolgimento complementare dei ruoli di Amministratore e Responsabile. L'Amministratore in questa fase ha dovuto scontrarsi con l'evidente carenza di documentazione ufficiale rendendo difficile l'attività iniziale di studio delle tecnologie e la formazione dei componenti del gruppo. I componenti, da parte loro, hanno cercato di supportare l'Amministratore nello svolgimento del suo compito effettuando in autonomia attività di test legate alle varie tecnologie coinvolte nel progetto.

B.2.2.2 Responsabile

Grazie al miglioramento dei rapporti con l'Amministratore e vista l'esperienza maturata nella prima fase del progetto, il Responsabile ha potuto svolgere il proprio ruolo con meno difficoltà. Il gruppo si è infatti sempre mostrato collaborativo e ha sempre portato a termine i propri compiti nei tempi previsti o con ritardi privi di incidenza negativa sul proseguimento delle attività. Inoltre, l'organizzazione di gruppi di lavoro più piccoli ha permesso anche di ridurre le tensioni tra i componenti.

B.2.2.3 Analista

Nel momento in cui il proponente ha richiesto una variazione ai requisiti iniziali, gli Analisti hanno dimostrato grande maturità nella gestione dei cambiamenti valutando con il Responsabile e l'Amministratore gli effetti di tale richiesta e attivandosi perché le modifiche non rallentassero le altre attività in corso.

B.2.2.4 Progettisti

I Progettisti hanno sottolineato l'importanza che l'incontro con il proponente ha avuto nella definizione della progettazione del PoC, indicando che ciò permetterà inoltre di facilitare la definizione del design di dettaglio nella prossima fase.

B.2.2.5 Programmatori

I Programmatori hanno riscontrato alcune difficoltà nell'utilizzo delle tecnologie e dei framework comunicando di conseguenza all'Amministratore che alcune versioni in utilizzo non erano completamente stabili o supportate a causa della natura open-source dei software utilizzati. Tuttavia, gli Amministratori si sono adoperati per ridurre al minimo tale problematica e hanno comunicato al Responsabile l'esigenza di accordarsi con il proponente in merito all'utilizzo di release più datate ma stabili. Infine, anche i Programmatori hanno espresso il proprio parere favorevole alla scelta del Responsabile di organizzare gruppi di lavoro contenuti anche nelle prossime fasi di progetto.

B.2.2.6 Verificatori

Niente da segnalare.

B.2.3 Valutazione sull'uso degli strumenti

Le problematiche riguardanti gli strumenti sono principalmente legate ad un erraneo utilizzo da parte dei componenti del gruppo dei sistemi di ticketing in utilizzo, sia su GitHub, sia su Microsoft Teams tramite il Planner. È stata quindi sottolineata l'importanza di mantenere costantemente aggiornata la dashboard e lo stato di avanzamento delle attività in modo che tutti i componenti del gruppo, in ogni momento, possano conoscere lo stato del prodotto e i compiti che gli altri membri del gruppo stanno svolgendo.

B.3 Periodo di Progettazione di dettaglio e codifica

B.3.1 Valutazione sull'organizzazione

Il gruppo sottolinea come la scelta, fatta nel periodo precedente, di suddividere il team in due gruppi rispettivamente responsabili dell'applicazione di addestramento e del plug-in abbia funzionato molto bene. Tale suddivisione è risultata particolarmente funzionale specialmente vista la situazione che si è creata nella quale tutto il lavoro collaborativo deve essere svolto telematicamente. Con questa modalità è infatti più facile coordinare le persone coinvolte in una data attività ed è stato possibile raggiungere un maggior numero di obiettivi rispetto a quelli previsti, dal punto di vista delle funzionalità del prodotto.

B.3.2 Valutazione sui ruoli

In questo periodo tutti i ruoli sono stati svolti con buoni risultati, sono emerse le propensioni dei singoli verso specifici ruoli e il buon clima di lavoro ha favorito la cooperazione e il raggiungimento degli obiettivi di periodo.

L'unico ruolo ad aver incontrato alcuni problemi è stato quello del Progettista. Le scarse conoscenze del gruppo in merito al mondo del Machine Learning ha richiesto un ulteriore approfondimento tecnologico con il Proponente per comprendere come gestire la funzionalità di previsione all'interno del plug-in. I tempi tecnici di organizzazione dell'incontro e la reperibilità del Proponente hanno quindi influito sul ritardo del completamento della progettazione. Allo stesso modo il ritardo di erogazione delle lezioni dovuto all'emergenza sanitaria in atto ha comportato ulteriori difficoltà al gruppo. I componenti del gruppo hanno infatti dovuto attendere che alcune lezioni venissero svolte per approfondirne i contenuti ed applicarli correttamente alla progettazione in atto.

B.3.3 Valutazione sull'uso degli strumenti

Il team evidenzia i buoni esiti ottenuti dall'utilizzo di Jest ed ESLint per le attività di verifica del software. ESLint è infatti facilmente integrabile con l'IDE che il gruppo utilizza, Visual Studio Code, permettendo di avere un feedback immediato riguardante eventuali errori individuati dall'analisi statica del codice sorgente. Per quanto riguarda Jest, nonostante i problemi di configurazione iniziale riscontrati con Grafana, si è rivelato anch'esso un ottimo strumento per eseguire test dinamici del software. Inoltre, questo strumento offre la possibilità di analizzare il livello di code coverage.

Infine, attraverso la configurazione di watchAll permette l'esecuzione continua dei test in modo non invasivo, ma fornendo allo sviluppatore feedback immediati, positivi o negativi, sullo stato del prodotto analizzato.

B.4 Periodo di Valutazione e collaudo

B.4.1 Valutazione conclusiva

Il team ha svolto un'analisi critica della propria esperienza di progetto dalla quale è emersa una condivisa consapevolezza di crescita personale e professionale. In particolare, è stato evidenziato come nel tempo siano maturate le capacità di organizzazione di tutti i membri del gruppo, così come una miglior attuazione dei compiti dei singoli ruoli svolti, che si sono tradotte in un aumento della produttività, specialmente nelle ultime fasi. Il gruppo ha inoltre sottolineato che le proprie competenze tecniche sono migliorate, grazie all'uso di tecnologie e strumenti che non erano mai stati utilizzati. I rischi riscontrati nel tempo e la capacità di affrontarli e risolverli ha anche incrementato l'abilità di problem solving dei singoli, i quali hanno compreso l'importanza della collaborazione e della condivisione per il superamento di situazioni critiche o di stallo.

In conclusione, il gruppo si ritiene maturato e ritiene che una parte fondamentale di questo processo di crescita sia da attribuire alle valutazioni periodiche del metodo di lavoro, svolte al termine di ogni periodo.