



ARGO

Piano di Qualifica

Gruppo Argo — Progetto ChatSQL

Informazioni sul documento

Versione	2.0.0
Approvazione	Sebastiano Lewental
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Argo



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Registro delle modifiche

Ver.	Data	Redazione	Verifica	Descrizione
2.0.0	2024-09-20	Sebastiano Lewental	Sebastiano Lewental	Approvazione e rilascio del documento
1.1.3	2024-09-19	Riccardo Cavalli	Tommaso Stocco	Aggiornamento grafici
1.1.2	2024-09-15	Tommaso Stocco	Riccardo Cavalli	Aggiornamento Indice Gulpease
1.1.1	2024-09-10	Riccardo Cavalli	Mattia Zecchinato, Marco Cristo, Raul Pianon	Registrazione esito test di accettazione
1.1.0	2024-09-05	Riccardo Cavalli	Marco Cristo	Revisione generale delle metriche
1.0.8	2024-09-05	Riccardo Cavalli	Raul Pianon	Aggiornamento metriche e grafici
1.0.7	2024-08-31	Riccardo Cavalli	Raul Pianon	Registrazione esito test di sistema
1.0.6	2024-08-28	Mattia Zecchinato, Marco Cristo	Riccardo Cavalli	Completata sezione test di integrazione
1.0.5	2024-08-24	Mattia Zecchinato, Marco Cristo	Riccardo Cavalli	Completata sezione test di unità
1.0.4	2024-08-22	Martina Dall'Amico, Riccardo Cavalli	Sebastiano Lewental, Tommaso Stocco, Mattia Zecchinato	Aggiornamento cruscotto e grafici
1.0.3	2024-08-19	Riccardo Cavalli	Mattia Zecchinato, Marco Cristo	Aggiunti test di unità front-end
1.0.2	2024-08-15	Riccardo Cavalli	Martina Dall'Amico, Sebastiano Lewental	Aggiunto test di unità per pulsante elimina chat
Continua nella prossima pagina				



Ver.	Data	Redazione	Verifica	Descrizione
1.0.1	2024-08-10	Riccardo Cavalli	Mattia Zecchinato, Tommaso Stocco	Aggiornamento cruscotto di valutazione della qualità
1.0.0	2024-07-25	Tommaso Stocco	Tommaso Stocco	Approvazione e rilascio del documento
0.2.1	2024-07-20	Tommaso Stocco, Riccardo Cavalli	Mattia Zecchinato	Allineamento grafici allo sprint 9
0.2.0	2024-07-12	Tommaso Stocco, Riccardo Cavalli	Mattia Zecchinato	Revisione generale del <i>Piano di Qualifica</i>
0.1.2	2024-07-05	Riccardo Cavalli, Mattia Zecchinato	Martina Dall'Amico, Marco Cristo	Inserimento grafico metrica EAC e descrizione dei grafici
0.1.1	2024-07-02	Tommaso Stocco	Riccardo Cavalli	Inserimento grafico metrica variazione di piano
0.1.0	2024-07-01	Riccardo Cavalli	Tommaso Stocco, Mattia Zecchinato	Revisione metriche e aggiornamento struttura del documento
0.0.7	2024-07-01	Riccardo Cavalli	Tommaso Stocco, Mattia Zecchinato	Stesura sezioni incomplete <i>Piano di Qualifica</i>
0.0.6	2024-06-29	Riccardo Cavalli	Tommaso Stocco, Mattia Zecchinato	Conversione in <i>LaTeX</i> sezione test
0.0.5	2024-06-28	Tommaso Stocco	Riccardo Cavalli	Aggiornamento cruscotto di qualità
0.0.4	2024-06-18	Raul Pianon, Riccardo Cavalli	Marco Cristo, Mattia Zecchinato, Tommaso Stocco	Inserimento grafici per le metriche
Continua nella prossima pagina				

Ver.	Data	Redazione	Verifica	Descrizione
0.0.3	2024-06-03	Sebastiano Lewental	Riccardo Cavalli, Raul Pianon, Marco Cristo	Aggiornamento metriche
0.0.2	2024-05-15	Martina Dall'Amico	Sebastiano Lewental	Inserimento tabelle delle metriche
0.0.1	2024-04-28	Riccardo Cavalli	Martina Dall'Amico, Mattia Zecchinato	Prima stesura del documento

Indice

1	Introduzione	8
1.1	Scopo del documento	8
1.2	Riferimenti	8
1.2.1	Riferimenti normativi	8
1.2.2	Riferimenti informativi	8
1.3	Glossario	11
1.4	Note organizzative	11
2	Obiettivi di qualità	12
2.1	Qualità di processo	12
2.2	Qualità di prodotto	13
2.2.1	Tracciamento delle metriche di prodotto	14
2.3	Panoramica degli obiettivi di qualità	15
2.3.1	Processi primari	15
2.3.1.1	Fornitura	15
2.3.1.2	Sviluppo	16
2.3.1.3	Codifica	16
2.3.2	Processi di supporto	16
2.3.2.1	Verifica	16
2.3.2.2	Documentazione	17
2.3.2.3	Assicurazione della qualità	17
2.3.3	Processi organizzativi	18
2.3.3.1	Gestione dei rischi	18
3	Verifica	19
3.1	Test di unità	19
3.2	Test di integrazione	36
3.3	Test di sistema	41
3.3.1	Tracciamento dei test di sistema	45
3.4	Test di accettazione	46
3.4.1	Tracciamento dei test di accettazione	51
3.5	Checklist	52
3.5.1	Struttura della documentazione	53
3.5.2	Errori formali	54
3.5.3	Analisi dei Requisiti	56
3.5.4	Codifica	57
4	Cruscotto di valutazione della qualità	58
4.1	M.PC.1 - Percentuale di metriche soddisfatte	58
4.2	M.PC.5 - EAC (Estimated at Completion)	60
4.3	M.PC.6 - Variazione del budget tra preventivo e consuntivo	61
4.4	M.PC.7 - Variazione del piano tra preventivo e consuntivo	62
4.5	M.PC.8 - Efficienza temporale	63
4.6	M.PC.9 - Frequenza di merge delle pull request	64
4.7	M.PC.10 - Indice di stabilità dei requisiti	66
4.8	M.PC.11 - Rischi inattesi	67
4.9	M.PC.12 - Efficacia delle contromisure nei rischi	68



4.10	M.PD.4 - Indice Gulpease	69
4.11	M.PD.1 - Percentuale di requisiti obbligatori soddisfatti	71
4.12	M.PD.2 - Percentuale di requisiti desiderabili soddisfatti	72
4.13	M.PD.3 - Percentuale di requisiti opzionali soddisfatti	73
4.14	M.PD.5 - Completezza descrittiva	74
4.15	M.PD.6 - Browser supportati	75
4.16	M.PD.11 - Code coverage	76
4.17	M.PD.12 - Adeguatezza delle funzioni sviluppate	77
4.18	M.PD.13 - Linee medie di codice per metodo	78
4.19	M.PD.14 - Complessità ciclomatica	79
4.20	M.PD.17 - Percentuale di test superati	80
4.21	M.PD.18 - Rimozione dei difetti	81
4.22	M.PD.19 - Tolleranza agli errori	82
4.23	M.PD.20 - Impatto delle modifiche	83
4.24	Altre metriche del codice	84

Elenco delle tabelle

2.1	Metriche di qualità di processo	12
2.2	Metriche di qualità di prodotto	13
2.3	Tracciamento metriche di prodotto	14
2.4	Metriche per obiettivo - Fornitura	15
2.5	Metriche per obiettivo - Sviluppo	16
2.6	Metriche per obiettivo - Codifica	16
2.7	Metriche per obiettivo - Verifica	17
2.8	Metriche per obiettivo - Documentazione	17
2.9	Metriche per obiettivo - Assicurazione della qualità	17
2.10	Metriche per obiettivo - Gestione dei rischi	18
3.1	Test di unità	19
3.2	Test di integrazione	37
3.3	Test di sistema	41
3.4	Tracciamento test di sistema	45
3.5	Test di accettazione	46
3.6	Tracciamento test di accettazione	51
3.7	Checklist - Struttura della documentazione	53
3.8	Checklist - Errori formali	54
3.9	Checklist - Analisi dei Requisiti	56
3.10	Checklist - Codifica	57
4.1	Tabella Indice Gulpease	70
4.2	Tabella dei browser supportati	75
4.3	Metriche del codice	84

Elenco delle figure

4.1	M.PC.1 – Percentuale di metriche soddisfatte	58
4.2	M.PC.5 – EAC (Estimated at Completion)	60
4.3	M.PC.6 – Variazione del budget tra preventivo e consuntivo	61
4.4	M.PC.7 – Variazione del piano tra preventivo e consuntivo	62
4.5	M.PC.8 – Efficienza temporale	63
4.6	M.PC.9 – Frequenza di merge delle pull request	64
4.7	M.PC.10 – Indice di stabilità dei requisiti	66
4.8	M.PC.11 – Rischi inattesi	67
4.9	M.PC.12 – Efficacia delle contromisure nei rischi	68
4.10	M.PD.4 – Indice Gulpease	69
4.11	M.PD.1 – Percentuale di requisiti obbligatori soddisfatti	71
4.12	M.PD.2 – Percentuale di requisiti desiderabili soddisfatti	72
4.13	M.PD.3 – Percentuale di requisiti opzionali soddisfatti	73
4.14	M.PD.5 – Completezza descrittiva	74
4.15	M.PD.11 – Code coverage	76
4.16	M.PD.12 – Adeguatezza delle funzioni sviluppate	77
4.17	M.PD.13 – Linee medie di codice per metodo	78
4.18	M.PD.14 – Complessità ciclomatica	79
4.19	M.PD.17 – Percentuale di test superati	80
4.20	M.PD.18 – Rimozione dei difetti	81
4.21	M.PD.19 – Tolleranza agli errori	82
4.22	M.PD.20 – Impatto delle modifiche	83

1 Introduzione

1.1 Scopo del documento

Lo scopo del *Piano di Qualifica* è delineare un insieme di indici di valutazione e validazione del progetto, assieme a delle metriche di qualità che il prodotto deve rispettare. Gli obiettivi di qualità devono essere chiari, quantificabili e conformi ai requisiti e alle aspettative del cliente. I parametri vengono fissati dal team sulla base di standard qualitativi e dell'esperienza acquisita nell'arco dello svolgimento del progetto. In linea con la dinamicità del *Piano di Qualifica*, i range possono essere aggiustati o migliorati. Per tale motivo, viene fornito un cruscotto di valutazione della qualità, che monitora la capacità del team di rispettare le metriche stabilite durante il progetto.

1.2 Riferimenti

1.2.1 Riferimenti normativi

- *Norme di Progetto v2.0.0*;
- Capitolato C9 - ChatSQL:
 - <https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C9.pdf>
(Ultimo accesso: 2024-09-20);
 - <https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C9p.pdf>
(Ultimo accesso: 2024-09-20).
- Slide PD2 - Corso di Ingegneria del Software - Regolamento del Progetto Didattico:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>
(Ultimo accesso: 2024-09-20).
- Standard ISO/IEC 9126:
https://it.wikipedia.org/wiki/ISO/IEC_9126
(Ultimo accesso: 2024-09-20);
- Standard ISO/IEC 9126 (inglese):
https://en.wikipedia.org/wiki/ISO/IEC_9126
(Ultimo accesso: 2024-09-20);
- Standard ISO/IEC 90003:
<https://cdn.standards.iteh.ai/samples/35867/36860aa4caba4c84b26051db5-76456d3/ISO-IEC-90003-2004.pdf>
(Ultimo accesso: 2024-09-20).

1.2.2 Riferimenti informativi

- *Analisi dei Requisiti v2.0.0*;
- Slide T7 - Corso di Ingegneria del Software - Qualità del software
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T7.pdf>
(Ultimo accesso: 2024-09-20);



- Slide T8 – Corso di Ingegneria del Software – Qualità di processo
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T8.pdf>
(Ultimo accesso: 2024-09-20);
- Slide T10 – Corso di Ingegneria del Software – Verifica e validazione: analisi statica
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T10.pdf>
(Ultimo accesso: 2024-09-20);
- Slide T11 – Corso di Ingegneria del Software – Verifica e validazione: analisi dinamica
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T11.pdf>
(Ultimo accesso: 2024-09-20);
- Panoramica generale sul test del software:
<https://www.ibm.com/it-it/topics/software-testing>
(Ultimo accesso: 2024-09-20);
- Esempi di metriche di prodotto:
http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf
(Ultimo accesso: 2024-09-20);
- Tipologie di test del software:
<https://www.atlassian.com/it/continuous-delivery/software-testing/types-of-software-testing>
(Ultimo accesso: 2024-09-20);
- Test di unità:
https://en.wikipedia.org/wiki/Unit_testing
(Ultimo accesso: 2024-09-20);
- Test di integrazione:
https://en.wikipedia.org/wiki/Integration_testing
(Ultimo accesso: 2024-09-20);
- Test di sistema:
<https://vitolavecchia.altervista.org/differenza-tra-system-testing-e-system-integration-testing>
(Ultimo accesso: 2024-09-20);
- Test di accettazione:
<https://vitolavecchia.altervista.org/tipologie-testing-software-test-di-accettazione>
(Ultimo accesso: 2024-09-20);
- Metriche del software:
<https://learn.microsoft.com/en-us/visualstudio/code-quality/code-metrics-values?view=vs-2022>
(Ultimo accesso: 2024-09-20);
- Elenco delle principali metriche di progetto:
https://it.wikipedia.org/wiki/Metriche_di_progetto
(Ultimo accesso: 2024-09-20);

- Usabilità dei siti web:
<https://usabile.it/392009-htm>
(Ultimo accesso: 2024-09-20);
- Indici di misurazione del software:
https://en.wikipedia.org/wiki/Software_metric
(Ultimo accesso: 2024-09-20);
- *Glossario v2.0.0*;
- *Piano di Progetto v2.0.0*;
- *Specifica Tecnica v1.0.0*;
- *Manuale Utente v1.0.0*;
- Verbali interni:
 - 2024-04-03;
 - 2024-04-10;
 - 2024-04-16;
 - 2024-04-20;
 - 2024-04-25;
 - 2024-05-02;
 - 2024-05-07;
 - 2024-05-16;
 - 2024-05-23;
 - 2024-05-28;
 - 2024-06-03;
 - 2024-06-14;
 - 2024-06-22;
 - 2024-07-06;
 - 2024-07-10;
 - 2024-07-18;
 - 2024-07-26;
 - 2024-08-01;
 - 2024-08-08;
 - 2024-08-14;
 - 2024-08-19;
 - 2024-08-27;
 - 2024-09-08;



- 2024-09-16.
- Verbali esterni:
 - 2024-04-09;
 - 2024-05-06;
 - 2024-05-22;
 - 2024-06-07;
 - 2024-07-09;
 - 2024-09-09.

1.3 Glossario

Allo scopo di evitare incomprensioni relative al linguaggio utilizzato nella documentazione di progetto, viene fornito un *Glossario*, nel quale ciascun termine è corredato da una spiegazione che mira a disambiguare il suo significato. I termini tecnici, gli acronimi e i vocaboli ritenuti ambigui vengono formattati in corsivo all'interno dei rispettivi documenti e marcati con una lettera _G in pedice. Tutte le ricorrenze di un termine definito nel *Glossario* subiscono la formattazione sopracitata.

1.4 Note organizzative

Il *Piano di Qualifica* è un documento dinamico; pertanto, la sua struttura e il suo contenuto sono soggetti a costanti aggiornamenti e miglioramenti.

2 Obiettivi di qualità

Questa sezione illustra i valori accettabili e ambiti delle metriche individuate dal team. Le metriche sono suddivise in:

- Metriche di processo;
- Metriche di prodotto.

La definizione di ciascuna metrica è riportata nel documento *Norme di Progetto v2.0.0*, §sezione Accertamento della qualità.

2.1 Qualità di processo

Le metriche di processo sono indicatori utilizzati per monitorare e valutare la qualità dei processi coinvolti nello sviluppo del software. Gli indici di misurazione individuati dal team contribuiscono al miglioramento della produttività e all'ottimizzazione delle procedure di gestione del progetto.

Tabella 2.1: Metriche di qualità di processo

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PC.1	Percentuale di metriche soddisfatte	$\geq 75\%$	100%
M.PC.2	AC (Actual cost)	≥ 0	$\leq \text{EAC}$
M.PC.3	EV (Earned Value)	≥ 0	$\leq \text{EAC}$
M.PC.4	PV (Planned Value)	≥ 0	$\leq \text{BAC}$
M.PC.5	EAC (Estimated at Completion)	$\pm 5\%$ rispetto al BAC	BAC
M.PC.6	Variazione del budget tra preventivo e consuntivo	$\pm 10\%$	$\pm 5\%$
M.PC.7	Variazione del piano tra preventivo e consuntivo	$\leq 15\%$	$\leq 5\%$
M.PC.8	Efficienza temporale	$\leq 180\%$	100%
M.PC.9	Frequenza di merge delle pull request	1 al giorno	2 al giorno
M.PC.10	Indice di stabilità dei requisiti	$\geq 70\%$	100%
M.PC.11	Rischi inattesi	≤ 2	0
Continua nella prossima pagina			

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PC.12	Efficacia delle contromisure nei rischi	$\geq 60\%$	100%

2.2 Qualità di prodotto

Le metriche di prodotto sono indicatori quantitativi e qualitativi utilizzati per valutare in modo obiettivo le caratteristiche del software. L'applicazione di queste metriche mira ad assicurare la conformità del software agli standard di qualità e ad aumentare il grado di soddisfazione del cliente.

Tabella 2.2: Metriche di qualità di prodotto

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PD.1	Requisiti obbligatori soddisfatti	$\geq 95\%$	100%
M.PD.2	Requisiti desiderabili soddisfatti	$\geq 30\%$	$\geq 60\%$
M.PD.3	Requisiti opzionali soddisfatti	$\geq 10\%$	$\geq 40\%$
M.PD.4	Indice Gulpease	≥ 50	≥ 80
M.PD.5	Completezza descrittiva	$\geq 90\%$	100%
M.PD.6	Browser supportati	$\geq 80\%$	100%
M.PD.7	Profondità (click necessari per reperire un'informazione)	4 click	2 click
M.PD.8	Ampiezza (opzioni nel menu di navigazione principale)	5 opzioni	3 opzioni
M.PD.9	Tempo di apprendimento	≤ 5 minuti	≤ 3 minuti
M.PD.10	Tempo di risposta	≤ 15 secondi	≤ 10 secondi
M.PD.11	Code coverage	$\geq 80\%$	$\geq 90\%$
M.PD.12	Adeguatezza delle funzioni sviluppate	$\geq 70\%$	$\geq 90\%$
Continua nella prossima pagina			

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PD.13	Linee medie di codice per metodo	≤ 30	≤ 15
M.PD.14	Complessità ciclomatica	≤ 10	≤ 5
M.PD.15	Duplicazione del codice	5% - 10%	3% - 5%
M.PD.16	Indice di manutenibilità	B	A
M.PD.17	Percentuale di test superati	$\geq 80\%$	100%
M.PD.18	Rimozione dei difetti	$\geq 70\%$	$\geq 90\%$
M.PD.19	Tolleranza agli errori	$\geq 70\%$	100%
M.PD.20	Impatto delle modifiche	$\leq 10\%$	$\leq 5\%$

2.2.1 Tracciamento delle metriche di prodotto

Tabella 2.3: Tracciamento metriche di prodotto

Caratteristica	Descrizione	Metriche
Funzionalità	Il software deve implementare i requisiti riportati nel documento di <i>Analisi dei Requisiti</i> .	M.PD.1, M.PD.2, M.PD.3, M.PD.12
Compatibilità	L'applicazione web deve essere compatibile con i seguenti browser: <ul style="list-style-type: none"> • Mozilla Firefox; • Google Chrome; • Safari; • Microsoft Edge; • Opera. 	M.PD.6
Continua nella prossima pagina		

Tabella 2.3: Tracciamento metriche di prodotto (continua)

Caratteristica	Caratteristica	Metriche
Usabilità	Il software deve facilitare l'interazione e il reperimento delle informazioni da parte dell'utente, senza provocare sovraccarico cognitivo.	M.PD.4, M.PD.5, M.PD.7, M.PD.8, M.PD.9
Efficienza	Il software deve fornire prestazioni adeguate in relazione alla quantità di risorse usate.	M.PD.10
Affidabilità	Il software deve rispettare le specifiche tecniche di funzionamento nel tempo.	M.PD.17, M.PD.18, M.PD.19
Manutenibilità	Il software deve poter essere modificato senza richiedere uno sforzo eccessivo in termini di tempo e costi.	M.PD.11, M.PD.13, M.PD.14, M.PD.15, M.PD.16, M.PD.20

2.3 Panoramica degli obiettivi di qualità

2.3.1 Processi primari

2.3.1.1 Fornitura

Tabella 2.4: Metriche per obiettivo - Fornitura

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PC.2	AC (Actual cost)	≥ 0	\leq EAC
M.PC.3	EV (Earned Value)	≥ 0	\leq EAC
M.PC.4	PV (Planned Value)	≥ 0	\leq BAC
M.PC.5	EAC (Estimated at Completion)	$\pm 5\%$ rispetto al BAC	BAC
M.PC.6	Variazione del budget tra preventivo e consuntivo	$\pm 10\%$	$\pm 5\%$
M.PC.7	Variazione del piano tra preventivo e consuntivo	$\leq 15\%$	$\leq 5\%$

2.3.1.2 Sviluppo

Tabella 2.5: Metriche per obiettivo - Sviluppo

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PC.8	Efficienza temporale	$\leq 180\%$	100%
M.PC.9	Frequenza di merge delle pull request	1 al giorno	2 al giorno
M.PC.10	Indice di stabilità dei requisiti	$\geq 70\%$	100%
M.PD.1	Requisiti obbligatori soddisfatti	$\geq 95\%$	100%
M.PD.2	Requisiti desiderabili soddisfatti	$\geq 30\%$	$\geq 60\%$
M.PD.3	Requisiti opzionali soddisfatti	$\geq 10\%$	$\geq 40\%$

2.3.1.3 Codifica

Tabella 2.6: Metriche per obiettivo - Codifica

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PD.11	Code coverage	$\geq 80\%$	$\geq 90\%$
M.PD.13	Linee medie di codice per metodo	≤ 30	≤ 15
M.PD.14	Complessità ciclomatica	≤ 10	≤ 5
M.PD.15	Duplicazione del codice	5% - 10%	3% - 5%
M.PD.16	Indice di manutenibilità	B	A

2.3.2 Processi di supporto

2.3.2.1 Verifica

Tabella 2.7: Metriche per obiettivo - Verifica

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PD.12	Adeguatezza delle funzioni sviluppate	$\geq 70\%$	$\geq 90\%$
M.PD.17	Percentuale di test superati	$\geq 80\%$	100%
M.PD.20	Impatto delle modifiche	$\leq 10\%$	$\leq 5\%$

2.3.2.2 Documentazione

Tabella 2.8: Metriche per obiettivo - Documentazione

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PD.4	Indice Gulpease	≥ 50	≥ 80
M.PD.5	Completezza descrittiva	$\geq 90\%$	100%

2.3.2.3 Assicurazione della qualità

Tabella 2.9: Metriche per obiettivo - Assicurazione della qualità

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PC.1	Percentuale di metriche soddisfatte	$\geq 75\%$	100%
M.PD.6	Browser supportati	$\geq 80\%$	100%
M.PD.7	Profondità (click necessari per reperire un'informazione)	4 click	2 click
M.PD.8	Ampiezza (opzioni nel menu di navigazione principale)	5 opzioni	3 opzioni
Continua nella prossima pagina			

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PD.9	Tempo di apprendimento	≤ 5 minuti	≤ 3 minuti
M.PD.10	Tempo di risposta	≤ 15 secondi	≤ 10 secondi
M.PD.18	Rimozione dei difetti	$\geq 70\%$	$\geq 90\%$
M.PD.19	Tolleranza agli errori	$\geq 70\%$	100%

2.3.3 Processi organizzativi

2.3.3.1 Gestione dei rischi

Tabella 2.10: Metriche per obiettivo - Gestione dei rischi

ID	Nome metrica	Valore tollerabile	Valore ambito
M.PC.11	Rischi inattesi	≤ 2	0
M.PC.12	Efficacia delle contromisure nei rischi	$\geq 60\%$	100%

3 Verifica

Il testing del software è un insieme di attività volte a garantire il soddisfacimento degli obiettivi di qualità. La fase di verifica può aiutare il team a identificare e risolvere con prontezza anomalie legate alle componenti software. Pertanto, il gruppo si impegna a eseguire i test contestualmente alle attività di sviluppo. Con questa procedura, il team si aspetta di ridurre l'impatto degli errori, garantendo il rispetto del budget e dei tempi previsti. Il gruppo ha individuato quattro classi di test finalizzate ad assicurare la correttezza, completezza e affidabilità del software:

- **Test di unità:** attività di verifica di singole *unità* del software;
- **Test di integrazione:** verificano che i diversi moduli, componenti o servizi utilizzati dall'applicazione funzionino in modo integrato;
- **Test di sistema:** controllano il comportamento del sistema nel suo complesso e verificano che l'applicazione funzioni secondo i requisiti specificati;
- **Test di accettazione:** sono test formali che precedono il rilascio del prodotto e valutano se l'applicazione è conforme alle aspettative del cliente.

3.1 Test di unità

Lo scopo dei test di unità è verificare il corretto funzionamento delle "unità software", ossia delle porzioni o segmenti (come una funzione, una classe o un componente) testabili in modo autonomo e isolato all'interno del sistema. Di seguito è riportato l'elenco dei test di unità:

Tabella 3.1: Test di unità

ID	Descrizione	Stato
TU.1	Verificare che il metodo "create" della classe "FileFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare correttamente un'istanza di "JsonFileAdapter" quando il tipo di file specificato nella configurazione è "json";• Deve sollevare un'eccezione "ValueError" se il tipo di file specificato nella configurazione è sconosciuto.	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.2	<p>Verificare che il metodo "validate" della classe "JsonSchemaValidatorAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve restituire "True" se lo schema è valido secondo il modello di confronto per la validazione; • Deve restituire "False" se lo schema non è valido secondo il modello di confronto per la validazione; • Deve sollevare un'eccezione se non viene trovato il file da validare. 	S
TU.3	<p>Verificare che il metodo "get_admin_by_username" della classe "SqlAlchemyAuthenticationRepositoryAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve recuperare correttamente un utente dal database in base allo username specificato, verificando che la query venga eseguita sul modello "Admins" applicando il filtro appropriato; • Deve restituire "None" se non esiste alcun utente associato allo username fornito. 	S
TU.4	<p>Verificare che il metodo "create_authentication_repository" della classe "SqlAlchemyDbManagerFactory" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve creare correttamente un'istanza di "SqlAlchemyAuthenticationRepositoryAdapter"; • Deve sollevare un'eccezione se si verifica un errore durante la creazione dell'istanza. 	S
Continua nella prossima pagina		



Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.5	Verificare che il metodo "create_dictionary_repository" della classe "SqlAlchemyDbManagerFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare correttamente un'istanza di "SqlAlchemyDictionaryRepositoryAdapter";• Deve sollevare un'eccezione se si verifica un errore durante la creazione dell'istanza;• Deve creare una nuova istanza ad ogni chiamata, verificando che più chiamate al metodo restituiscano istanze distinte.	S
TU.6	Verificare che l'inizializzazione della classe "SqlAlchemyDbManagerFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve chiamare il metodo "create_all" per garantire che tutte le tabelle del database siano create.	S
TU.7	Verificare che il metodo "create_dictionary" della classe "SqlAlchemyDictionaryRepositoryAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare un nuovo dizionario e salvarlo nel database, verificando che le proprietà del dizionario siano impostate correttamente;• Deve chiamare le operazioni di add, commit e refresh durante l'inserimento;• Deve sollevare un'eccezione se si verifica un errore durante il commit del dizionario.	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.8	<p>Verificare che il metodo "update_dictionary" della classe "SqlAlchemyDictionaryRepositoryAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve aggiornare correttamente le proprietà di un dizionario esistente e salvare le modifiche nel database; • Deve chiamare le operazioni di commit e refresh durante l'aggiornamento; • Deve sollevare un'eccezione se si verifica un errore durante il commit delle modifiche; • Deve restituire "None" se si tenta di aggiornare un dizionario inesistente. 	S
TU.9	<p>Verificare che il metodo "delete_dictionary" della classe "SqlAlchemyDictionaryRepositoryAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve eliminare correttamente un dizionario dal database; • Deve chiamare le operazioni di delete e commit durante l'eliminazione; • Non deve chiamare le operazioni di delete e commit se si tenta di eliminare un dizionario inesistente; • Deve sollevare un'eccezione se si verifica un errore durante l'eliminazione del dizionario. 	S
TU.10	<p>Verificare che il metodo "get_all_dictionaries" della classe "SqlAlchemyDictionaryRepositoryAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve restituire correttamente tutti i dizionari presenti nel database, verificando che la query venga eseguita sul modello "Dictionaries"; • Deve chiamare il metodo "all" durante il recupero dei dizionari. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.11	Verificare che il metodo "get_dictionary_by_id" della classe "SqlAlchemyDictionaryRepositoryAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve restituire correttamente un dizionario in base all'ID fornito, verificando che la query venga eseguita sul modello "Dictionaries" applicando il filtro appropriato.	S
TU.12	Verificare che il metodo "get_dictionary_by_name" della classe "SqlAlchemyDictionaryRepositoryAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve restituire correttamente un dizionario in base al nome fornito, verificando che la query venga eseguita sul modello "Dictionaries" applicando il filtro appropriato.	S
TU.13	Verificare che il metodo "create" della classe "DbManagerFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare correttamente un'istanza di "SqlAlchemyDbManagerFactory" quando il tipo di database manager specificato nella configurazione è "sqlalchemy";• Deve sollevare un'eccezione "ValueError" se il tipo di database manager specificato nella configurazione è sconosciuto.• Deve sollevare un'eccezione "ValueError" se manca la chiave "db_manager" nella configurazione.	S
Continua nella prossima pagina		



Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.14	Verificare che il metodo "prompt_generator" della classe "TxtaiPromptManagerAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve generare correttamente il <i>prompt_e</i> includendo la richiesta dell'Utente, i metadati dello schema, la lingua e il <i>DBMS_e</i>;• Deve restituire un prompt vuoto se non vengono trovati risultati rilevanti;• Deve restituire il <i>log_e</i> se la modalità di <i>debug_e</i> è attiva;• Deve gestire correttamente il caso in cui la richiesta dell'Utente è vuota;• Deve gestire correttamente gli errori durante l'estrazione dei metadati dallo schema.	S
TU.15	Verificare che il metodo "get_index_manager" della classe "TxtaiPromptManagerAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve restituire correttamente l'istanza del gestore di indici associata alla classe.	S
TU.16	Verificare che il metodo "get_embeddings" della classe "TxtaiIndexManagerAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve restituire un'istanza della classe "Embeddings".	S
TU.17	Verificare che il metodo "create_or_load_index" della classe "TxtaiIndexManagerAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve ripristinare un <i>indice_e</i> e restituire "False" se il dizionario ha già un indice associato;• Deve creare un nuovo indice e restituire "True" se il dizionario non ha un indice associato.	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.18	<p>Verificare che il metodo "create_index" della classe "TxtailIndexManagerAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve creare un indice utilizzando i documenti estratti e salvarlo se "save_index" è impostato a "True"; • Deve creare un indice senza salvarlo se "save_index" è impostato a "False"; • Deve creare un indice anche se la lista dei documenti è vuota e salvarlo se "save_index" è impostato a "True"; • Deve sollevare un'eccezione se si verifica un errore durante l'indicizzazione. 	S
TU.19	<p>Verificare che il metodo "save_index" della classe "TxtailIndexManagerAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve salvare l'indice utilizzando il percorso corretto; • Deve sollevare un'eccezione se si verifica un errore durante il salvataggio dell'indice. 	S
TU.20	<p>Verificare che il metodo "load_index" della classe "TxtailIndexManagerAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve ripristinare l'indice utilizzando il percorso corretto. 	S
TU.21	<p>Verificare che il metodo "delete_index" della classe "TxtailIndexManagerAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve eliminare correttamente la directory dell'indice; • Non deve tentare di eliminare la directory dell'indice se quest'ultima non esiste. 	S
Continua nella prossima pagina		



Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.22	Verificare che il metodo "create_index_manager" della classe "TxtaiEmbeddingsManagerFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare correttamente un'istanza di "TxtaiIndexManagerAdapter";• Deve creare correttamente un'istanza di "TxtaiIndexManagerAdapter" anche se manca la chiave "txtai" nella configurazione.	S
TU.23	Verificare che il metodo "create_prompt_manager_with_dependencies" della classe "TxtaiEmbeddingsManagerFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare un'istanza di "TxtaiPromptManagerAdapter" quando viene fornita un'istanza di "TxtaiIndexManagerAdapter" come dipendenza.	S
TU.24	Verificare che il metodo "create_prompt_manager" della classe "TxtaiEmbeddingsManagerFactory" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve creare un'istanza di "TxtaiPromptManagerAdapter", gestendo internamente la creazione di un'istanza di "TxtaiIndexManagerAdapter".	S
TU.25	Verificare che il metodo "semantic_search" della classe "TxtaiSearchAlgorithmAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Deve eseguire correttamente una ricerca semantica tramite l'index manager e restituire i risultati pertinenti;• Deve generare il log corretto quando la modalità di debug è attiva, includendo i dettagli sull'importanza dei termini;• Deve restituire i risultati della ricerca e il contenuto del log in modo appropriato;• Deve restituire una lista vuota se la ricerca semantica non produce risultati.	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.26	<p>Verificare che il metodo "search_filtering" della classe "TxtaiSearchAlgorithmAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve filtrare correttamente i risultati della ricerca basandosi sui punteggi ("max_score"); • Non deve generare il log quando il logging è disattivato; • Deve gestire correttamente una lista vuota di tuple. 	S
TU.27	<p>Verificare che il metodo "semantic_search_log" della classe "TxtaiSearchAlgorithmAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve generare correttamente il log per una ricerca semantica, riportando i dettagli e i termini rilevanti per ogni descrizione; • Deve gestire correttamente la generazione del log quando non vengono trovati risultati. 	S
TU.28	<p>Verificare che il metodo "search_filtering_log" della classe "TxtaiSearchAlgorithmAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve generare un log che indichi quali tabelle vengono mantenute e quali vengono scartate; • Deve specificare il motivo per cui alcune tabelle vengono scartate (es.: differenza di punteggio > 0.2); • Deve gestire correttamente una lista vuota di tuple. 	S
TU.29	<p>Verificare che il metodo "get_debug_header" della classe "TxtaiSearchAlgorithmAdapter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve generare correttamente un'intestazione di debug con il livello di log e il sistema specificato; • Deve formattare l'intestazione secondo lo schema "[SystemName] [LogLevel]". 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.30	Verificare che il metodo "save" della classe "JsonFileAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve aprire un file con permessi di scrittura; • Deve eseguire almeno un'operazione di scrittura sul file aperto. 	S
TU.31	Verificare che il metodo "load" della classe "JsonFileAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire il percorso del file. 	S
TU.32	Verificare che il metodo "delete" della classe "JsonFileAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve controllare l'esistenza del file; • Deve eseguire l'operazione di eliminazione del file. 	S
TU.33	Verificare che il metodo "get_preview" della classe "JsonFileAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire un oggetto che rappresenti lo schema del file; • Deve restituire "None" se lo schema del file non è presente. 	S
TU.34	Verificare che il metodo "extract_index_metadata" della classe "JsonFileAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire i metadati del dizionario per l'indicizzazione. 	S
TU.35	Verificare che il metodo "extract_schema_metadata" della classe "JsonFileAdapter" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire i metadati del dizionario in forma di prompt. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.36	<p>Verificare che il componente "AppLogo" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il logo deve essere renderizzato con il percorso dell'immagine corretto; • L'altezza e la larghezza dell'immagine devono essere impostate con valori predefiniti se non vengono passate come props; • Il colore dell'immagine deve cambiare in base al tema selezionato. 	S
TU.37	<p>Verificare che il componente "AppMenu" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il menu dell'Utente deve avere avere una o più opzioni; • Il menu del Tecnico deve avere due o più opzioni. 	S
TU.38	<p>Verificare che il componente "AppMenuitem" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • La voce di menu deve contenere l'etichetta corretta; • La voce di menu non deve essere attiva se non è stata selezionata; • Una volta selezionata, la voce di menu deve essere attiva; • La voce di menu può essere la radice di un menu; • La voce di menu può contenere un sotto-menu; • La voce di menu può aprire o chiudere un sotto-menu; • L'icona "submenu-toggler" deve essere visibile se la voce di menu contiene un sotto-menu. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.39	<p>Verificare che il componente "AppTopbar" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il pulsante di login deve essere visibile se l'Utente non ha effettuato l'accesso; • Il pulsante di logout deve essere visibile se l'Utente ha effettuato l'accesso; • Quando viene cliccato il pulsante di logout, la funzione di logout deve essere chiamata; • L'option menu deve essere aperto al clic dell'apposito pulsante; • L'option menu deve essere chiuso quando l'Utente clicca al di fuori di esso. 	S
TU.40	<p>Verificare che il componente "AppFooter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il footer deve essere renderizzato correttamente; • All'interno del footer deve essere visualizzata la versione corretta dell'applicazione. 	S
TU.41	<p>Verificare che il componente "ConfigSidebar" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • La sezione per modificare la scala deve essere visibile; • La sezione per modificare il tema deve essere visibile; • La sezione per modificare la lingua dell'interfaccia deve essere visibile; • La scala deve essere diminuita al clic dell'apposito pulsante; • La scala deve essere aumentata al clic dell'apposito pulsante; • Il pulsante per diminuire la scala deve essere disabilitato se la scala è al minimo; • Il pulsante per aumentare la scala deve essere disabilitato se la scala è al massimo; • Il contenuto del localStorage deve essere aggiornato al cambio del tema; • Il contenuto del localStorage deve essere aggiornato al cambio della lingua. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.42	<p>Verificare che il componente "MenuSidebar" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il pulsante per chiudere la sidebar deve essere nascosto su schermi di grandi dimensioni; • Il pulsante per chiudere la sidebar deve essere visibile su schermi di piccole dimensioni. 	S
TU.43	<p>Verificare che il componente "LoginDialog" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il dialog deve essere chiuso al clic dell'apposito pulsante; • La funzione "messageSuccess" deve essere chiamata se il login viene effettuato con successo; • Una volta completato il login, il dialog deve essere chiuso; • Il contenuto del localStorage deve essere aggiornato se il login viene effettuato con successo; • Una volta completato il login, il form deve essere resettato; • La funzione "messageError" deve essere chiamata se il login fallisce. 	S
TU.44	<p>Verificare che il componente "DictPreview" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • L'anteprima del dizionario dati deve essere nascosta se la variabile detailsVisible è uguale a "false"; • Se è visibile, l'anteprima deve mostrare i dati corretti; • Lo stato di espansione dell'anteprima deve essere attivato/disattivato al clic dell'apposito pulsante; • Quando viene cliccato il pulsante di chiusura, il componente deve inviare un segnale. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.45	<p>Verificare che il componente "ChatMessage" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il messaggio inviato dall'Utente deve essere visualizzato correttamente; • Il messaggio inviato dal ChatBOT deve essere visualizzato correttamente; • Se il messaggio inviato dal ChatBOT è "null", il sistema deve mostrare un avviso all'Utente; • I pulsanti di azione devono essere nascosti se il messaggio è stato inviato dall'Utente; • La funzione "Copia negli appunti" deve essere chiamata con i parametri corretti al clic dell'apposito pulsante; • Il pulsante di debug deve essere nascosto se l'Utente non ha effettuato il login; • La funzione "Apri debug modal" deve essere chiamata con i parametri corretti al clic dell'apposito pulsante. 	S
TU.46	<p>Verificare che il componente "DebugMessage" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il componente deve visualizzare il messaggio di debug correttamente; • Quando viene cliccato il pulsante "Scarica file", la funzione di download deve essere chiamata con i parametri corretti. 	S
TU.47	<p>Verificare che il componente "ChatDeleteBtn" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il pulsante deve essere disabilitato se non è presente alcun messaggio nella chat; • Il pulsante deve essere disabilitato se il ChatBOT sta elaborando un nuovo messaggio; • Quando viene cliccato, il pulsante deve inviare un segnale. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.48	<p>Verificare che il componente "CreateUpdateDictionaryModal" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il form di creazione e modifica deve essere visualizzato correttamente; • Il pulsante di invio deve essere disabilitato se il form è incompleto; • La funzione "messageSuccess" deve essere chiamata se l'inserimento viene effettuato con successo; • La funzione "messageError" deve essere chiamata se l'inserimento fallisce; • Il pulsante di invio deve essere disabilitato se il nome del dizionario dati contiene caratteri non supportati; • Il pulsante di invio deve essere disabilitato se il formato del file non è valido; • La funzione "messageSuccess" deve essere chiamata se l'aggiornamento dei metadati viene effettuato con successo; • La funzione "messageError" deve essere chiamata se l'aggiornamento dei metadati fallisce; • La funzione "messageSuccess" deve essere chiamata se l'aggiornamento del file viene effettuato con successo; • La funzione "messageError" deve essere chiamata se l'aggiornamento del file fallisce; • L'upload del file deve essere disabilitato se un file è già stato caricato; • Il file selezionato deve essere rimosso al clic dell'apposito pulsante. 	S
TU.49	<p>Verificare che il componente "AppLayout" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il layout dell'applicazione deve essere renderizzato correttamente; • Il wrapper globale deve contenere le classi corrette sulla base delle impostazioni e dello stato del layout. 	S
Continua nella prossima pagina		



Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.50	<p>Verificare che il componente "DictionariesListView" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none">• La lista dei dizionari dati deve essere visualizzata correttamente;• Se la lista dei dizionari è vuota, il sistema deve mostrare un messaggio esplicativo;• I risultati della ricerca dei dizionari per nome devono essere visualizzati correttamente;• Se la ricerca dei dizionari per nome non produce risultati, il sistema deve mostrare un messaggio esplicativo;• I risultati della ricerca dei dizionari per descrizione devono essere visualizzati correttamente;• Se la ricerca dei dizionari per descrizione non produce risultati, il sistema deve mostrare un messaggio esplicativo;• La funzione "Apri dictionary modal" deve essere chiamata con i parametri corretti al clic del pulsante di inserimento;• La funzione "Apri dictionary modal" deve essere chiamata con i parametri corretti al clic del pulsante di modifica dei metadati;• La funzione "Apri dictionary modal" deve essere chiamata con i parametri corretti al clic del pulsante di modifica del file;• Quando viene cliccato il pulsante "Scarica file", la funzione di download deve essere chiamata con i parametri corretti;• La funzione "messageSuccess" deve essere chiamata se l'eliminazione viene effettuata con successo;• La funzione "messageError" deve essere chiamata se l'eliminazione fallisce.	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.51	<p>Verificare che il componente "ChatView" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il contenuto della chat deve essere renderizzato correttamente; • Il form di selezione pre-richiesta deve essere attivato/disattivato al click dell'apposito pulsante; • Nella chat deve essere visualizzato il dizionario dati attivo con la relativa estensione; • Quando un dizionario viene selezionato, il contenuto del localStorage deve essere aggiornato; • Il pulsante di invio deve essere disattivato se non è stato selezionato alcun dizionario; • Il pulsante di invio deve essere disattivato se non è stata inserita alcuna richiesta; • La richiesta deve poter essere inviata premendo il tasto "Invio"; • La funzione di generazione del prompt deve essere chiamata con i parametri corretti al clic del pulsante di invio; • Quando il ChatBOT ritorna una risposta, il contenuto del sessionStorage deve essere aggiornato; • Quando viene inviata una richiesta, il campo di testo deve essere resettato; • La funzione "messageError" deve essere chiamata se la generazione del prompt fallisce; • Il pulsante "Scroll to bottom" deve essere visualizzato quando l'Utente scorre i messaggi; • L'ultimo messaggio della chat deve essere visualizzato se viene cliccato il pulsante "Scroll to bottom". 	S
TU.52	<p>Verificare che il metodo "logout" della classe "auth.service" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il metodo deve rimuovere il token di autenticazione dal localStorage; • Il metodo deve inviare un segnale di notifica. 	S
Continua nella prossima pagina		

Tabella 3.1: Test di unità (continua)

ID	Descrizione	Stato
TU.53	Verificare che il metodo "isLogged" della classe "auth.service" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Il metodo deve restituire "false" se il token di autenticazione non è presente;• Il metodo deve restituire "false" se il token è scaduto;• Il metodo deve restituire "true" se il token è valido;• Il metodo deve restituire "false" se la data di scadenza non è definita.	S
TU.54	Verificare che il metodo "downloadFile" della classe "utils.service" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Il metodo deve creare un collegamento per il download del file;• Il metodo deve attivare il download cliccando sul collegamento.	S
TU.55	Verificare che il metodo "stringToSnakeCase" della classe "utils.service" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Il metodo deve convertire una stringa in snake_case.	S
TU.56	Verificare che il metodo "addCapitalizeValues" della classe "utils.service" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Il metodo deve aggiungere chiavi in maiuscolo a un determinato oggetto.	S
TU.57	Verificare che il metodo "capitalizeString" della classe "utils.service" soddisfi le seguenti condizioni: <ul style="list-style-type: none">• Il metodo deve convertire in maiuscolo la prima lettera di una stringa.	S

3.2 Test di integrazione

Lo scopo dei test di integrazione è rilevare difetti di design o carenze nei test di unità, verificando che i moduli o componenti del sistema funzionino correttamente quando combinati tra loro.

Tabella 3.2: Test di integrazione

ID	Descrizione	Stato
TI.1	Verificare che il metodo "login" della classe "AuthenticationService" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire lo status "NOT_FOUND" se l'Utente non è registrato nel sistema; • Deve restituire lo status "BAD_CREDENTIAL" se la password fornita è errata; • Deve restituire lo status "OK" e un token di autenticazione quando il login viene effettuato con successo. 	S
TI.2	Verificare che il metodo "get_dictionary_list" della classe "DictionaryService" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire l'elenco completo dei dizionari. 	S
TI.3	Verificare che il metodo "get_dictionary_by_id" della classe "DictionaryService" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire il dizionario associato all'ID fornito; • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito. 	S
TI.4	Verificare che il metodo "get_dictionary_file" della classe "DictionaryService" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire il percorso del file relativo al dizionario specificato; • Deve restituire "None" se non esiste alcun dizionario associato all'ID fornito. 	S
TI.5	Verificare che il metodo "get_dictionary_preview" della classe "DictionaryService" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Deve restituire un'anteprima del dizionario associato all'ID fornito; • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito. 	S
Continua nella prossima pagina		

Tabella 3.2: Test di integrazione (continua)

ID	Descrizione	Stato
TI.6	<p>Verificare che il metodo "create_dictionary" della classe "DictionaryService" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve inserire un nuovo dizionario nel sistema; • Deve restituire lo status "CONFLICT" se il nome del dizionario non è univoco; • Deve restituire lo status "BAD_REQUEST" se i dati forniti sono incompleti o non validi. 	S
TI.7	<p>Verificare che il metodo "update_dictionary_metadata" della classe "DictionaryService" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve aggiornare correttamente i metadati del dizionario specificato; • Deve restituire lo status "CONFLICT" se il nome del dizionario non è univoco; • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito. 	S
TI.8	<p>Verificare che il metodo "update_dictionary_file" della classe "DictionaryService" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve aggiornare correttamente il file relativo al dizionario specificato; • Deve restituire lo status "BAD_REQUEST" se il contenuto del file è invalido; • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito. 	S
TI.9	<p>Verificare che il metodo "delete_dictionary" della classe "DictionaryService" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve eliminare correttamente il dizionario specificato; • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito. 	S
Continua nella prossima pagina		

Tabella 3.2: Test di integrazione (continua)

ID	Descrizione	Stato
TI.10	<p>Verificare che il metodo "generate_prompt" della classe "PromptManagerService" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito; • Deve restituire lo status "BAD_REQUEST" se la richiesta è incompleta; • Deve generare correttamente il <i>prompt_g</i> quando viene fornita una richiesta idonea. 	S
TI.11	<p>Verificare che il metodo "generate_prompt_with_debug" della classe "PromptManagerService" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Deve restituire lo status "NOT_FOUND" se non esiste alcun dizionario associato all'ID fornito; • Deve restituire lo status "BAD_REQUEST" se la richiesta è incompleta; • Deve generare correttamente il prompt e includere informazioni di <i>debug_g</i> relative ad esso quando viene fornita una richiesta idonea. 	S
TI.12	<p>Verificare che il componente "AppMenu" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • L'Utente deve poter accedere alle seguenti pagine: <ul style="list-style-type: none"> – Chat. • Il Tecnico deve poter accedere alle seguenti pagine: <ul style="list-style-type: none"> – Chat; – Gestione dizionari dati_g. 	S
TI.13	<p>Verificare che il componente "AppTopbar" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il logo deve essere renderizzato correttamente. 	S
TI.14	<p>Verificare che il componente "AppFooter" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • Il logo deve essere renderizzato correttamente. 	S
Continua nella prossima pagina		

Tabella 3.2: Test di integrazione (continua)

ID	Descrizione	Stato
TI.15	Verificare che il componente "MenuSidebar" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Il menu di navigazione principale deve essere visibile; • Il footer deve essere visibile. 	S
TI.16	Verificare che il componente "StringDataModal" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Il messaggio di debug deve essere visualizzato correttamente quando la variabile StringData è impostata; • Il messaggio di debug non deve essere renderizzato se la variabile StringData è vuota. 	S
TI.17	Verificare che il componente "AppLayout" soddisfi le seguenti condizioni: <ul style="list-style-type: none"> • Il dialog di login deve essere aperto/chiuso al clic dell'apposito pulsante; • La barra laterale delle impostazioni deve essere aperta/chiusa al clic dell'apposito pulsante; • La barra laterale delle impostazioni deve essere chiusa se l'Utente clicca al di fuori di essa; • La barra laterale del menu di navigazione deve essere aperta/chiusa al clic dell'apposito pulsante; • La barra laterale del menu di navigazione deve essere chiusa se l'Utente clicca al di fuori di essa. 	S
Continua nella prossima pagina		

Tabella 3.2: Test di integrazione (continua)

ID	Descrizione	Stato
TI.18	<p>Verificare che il componente "ChatView" soddisfi le seguenti condizioni:</p> <ul style="list-style-type: none"> • L'anteprima del dizionario dati selezionato deve essere visualizzata al clic dell'apposito pulsante; • L'anteprima del dizionario deve essere nascosta quando viene inviato un messaggio; • Quando viene aperta l'anteprima del dizionario, i messaggi della chat devono essere nascosti; • L'anteprima del dizionario deve essere nascosta al clic dell'apposito pulsante; • Quando viene chiusa l'anteprima del dizionario, i messaggi della chat devono essere resi nuovamente visibili; • Il pulsante di debug deve essere visualizzato se la richiesta è stata inviata dal Tecnico; • I messaggi della chat devono essere cancellati al clic dell'apposito pulsante. 	S

3.3 Test di sistema

I test di sistema devono assicurare una completa copertura dei requisiti concordati con la *Proponente*, e/o specificati nel documento di *Analisi dei Requisiti*. Di seguito è riportato l'elenco dei test di sistema:

Tabella 3.3: Test di sistema

ID	Descrizione	Stato
TS.1	Verificare che l'Utente possa effettuare il login.	S
TS.2	Verificare che l'Utente visualizzi un errore qualora inserisca delle credenziali errate in fase di autenticazione.	S
TS.3	Verificare che il Tecnico possa inserire un nuovo <i>dizionario dati</i> nel sistema.	S
TS.4	Verificare che il Tecnico possa modificare il nome di un dizionario dati.	S
TS.5	Verificare che il Tecnico possa modificare la descrizione di un dizionario dati.	S
Continua nella prossima pagina		

Tabella 3.3: Test di sistema (continua)

ID	Descrizione	Stato
TS.6	Verificare che il Tecnico possa modificare il file di configurazione di un dizionario dati.	S
TS.7	Verificare che il Tecnico visualizzi un errore nel caso in cui il nome del dizionario dati contenga caratteri non supportati.	S
TS.8	Verificare che il sistema restituisca un messaggio di errore qualora il Tecnico inserisca un nome già esistente per il dizionario dati.	S
TS.9	Verificare che il sistema restituisca un messaggio di errore qualora il Tecnico inserisca un dizionario dati con una dimensione superiore a 1 MB.	S
TS.10	Verificare che il Tecnico visualizzi un errore nel caso in cui l'estensione del file caricato sia diversa da <i>JSON_e</i> .	S
TS.11	Verificare che il sistema restituisca un messaggio di errore qualora il Tecnico inserisca un dizionario dati non conforme allo schema predefinito.	S
TS.12	Verificare che il Tecnico possa eliminare un dizionario dati dal sistema.	S
TS.13	Verificare che il Tecnico possa scaricare un dizionario dati.	S
TS.14	Verificare che l'Utente possa accedere alla chat e inserire un messaggio nella maschera di richiesta.	S
TS.15	Verificare che l'Utente possa selezionare un dizionario dati e renderlo operativo nel sistema.	S
TS.16	Verificare che l'Utente possa visualizzare un'anteprima del dizionario dati selezionato. Il sistema deve mostrare le seguenti informazioni: <ul style="list-style-type: none"> • Nome del database; • Descrizione del database; • Lista delle tabelle del database. Per ciascuna tabella devono essere riportate le seguenti informazioni: <ul style="list-style-type: none"> - Nome della tabella; - Descrizione della tabella. 	S
TS.17	Verificare che l'Utente possa inviare una richiesta al ChatBOT.	S
Continua nella prossima pagina		

Tabella 3.3: Test di sistema (continua)

ID	Descrizione	Stato
TS.18	Verificare che il sistema restituisca un avviso qualora l'Utente inserisca una richiesta ritenuta non idonea dal modello di $AI_{\mathcal{G}}$.	S
TS.19	Verificare che il sistema restituisca un messaggio di errore nel caso in cui la generazione del $prompt_{\mathcal{G}}$ venga interrotta senza preavviso.	S
TS.20	Verificare che l'Utente possa visualizzare correttamente il prompt generato. Il prompt deve contenere le seguenti informazioni: <ul style="list-style-type: none"> • Lista delle tabelle pertinenti. Ogni tabella deve essere corredata da: <ul style="list-style-type: none"> – Schema della tabella: composto dal nome della tabella e da una lista di colonne, a loro volte organizzate per nome e tipo (es.: integer, string); – Chiave primaria; – Descrizione della tabella; – Descrizione delle colonne della tabella; – Chiavi esterne. • $DBMS_{\mathcal{G}}$ di riferimento; • Lingua di riferimento; • Richiesta in linguaggio naturale. 	S
TS.21	Verificare che il Tecnico possa effettuare il logout per terminare la sessione corrente.	S
TS.22	Verificare che l'Utente possa copiare il contenuto del prompt generato.	S
TS.23	Verificare che il sistema generi un $log_{\mathcal{G}}$ se la richiesta viene inviata dal Tecnico.	S
TS.24	Verificare che il Tecnico possa scaricare un file di log contenente il $debug_{\mathcal{G}}$ del prompt.	S
TS.25	Verificare che l'Utente possa eliminare la cronologia della chat.	S
TS.26	Verificare che il Tecnico possa visualizzare la lista dei dizionari dati con le relative informazioni.	S
TS.27	Verificare che il sistema di generazione del prompt supporti richieste in lingue diverse dall'inglese.	S
Continua nella prossima pagina		

Tabella 3.3: Test di sistema (continua)

ID	Descrizione	Stato
TS.28	Verificare che l'Utente possa cambiare il tema dell'interfaccia da chiaro a scuro.	S
TS.29	Verificare che l'Utente possa aumentare la scala dell'interfaccia.	S
TS.30	Verificare che l'Utente possa diminuire la scala dell'interfaccia.	S
TS.31	Verificare che l'Utente possa cambiare la lingua dell'interfaccia da italiano a inglese.	S

3.3.1 Tracciamento dei test di sistema

Tabella 3.4: Tracciamento test di sistema

ID	Requisito
TS.1	RF.O.1, RF.O.1.1, RF.O.1.2
TS.2	R.F.O.2
TS.3	RF.O.13, RF.O.15, RF.O.16, RF.O.17
TS.4	RF.O.29
TS.5	RF.O.30
TS.6	RF.O.15, RF.O.20
TS.7	RF.O.31, RF.O.31.1
TS.8	RF.O.31, RF.O.31.2
TS.9	RF.O.33
TS.10	RF.O.28
TS.11	RF.O.34
TS.12	RF.O.18
TS.13	RF.O.38
TS.14	RF.O.5
TS.15	RF.O.4
TS.16	RF.O.14, RF.O.14.1, RF.O.14.2, RF.O.14.3, RF.O.14.3.1, RF.O.14.3.1.1, RF.O.14.3.1.2
TS.17	RF.O.3
TS.18	RF.O.6
TS.19	RF.O.11
TS.20	RF.O.35
TS.21	RF.O.12
TS.22	RF.O.8
TS.23	RF.D.46
TS.24	RF.O.23
TS.25	RF.O.27
TS.26	RF.O.9, RF.O.9.1, RF.O.10, RF.O.10.1, RF.O.10.2, RF.O.10.3
TS.27	RF.OP.52
Continua nella prossima pagina	

Tabella 3.4: Tracciamento test di sistema (continua)

ID	Requisito
TS.28	RF.O.60
TS.29	RF.O.61
TS.30	RF.O.61
TS.31	RF.D.62

3.4 Test di accettazione

L'obiettivo dei test di accettazione è verificare se il sistema soddisfa le aspettative del Committente e del Proponente. I test di accettazione determinano se il software è pronto per essere rilasciato, e pertanto richiedono un focus sul comportamento degli utenti. Di seguito è riportato l'elenco dei test di accettazione:

Tabella 3.5: Test di accettazione

ID	Descrizione	Stato
TA.1	Verificare che l'Utente possa effettuare il login: 1. Avviare la procedura di autenticazione (da qualsiasi pagina); 2. Inserire uno username; 3. Inserire una password; 4. Richiedere l'accesso; 5. Visualizzare un messaggio di conferma una volta effettuato il login.	S
Continua nella prossima pagina		

Tabella 3.5: Test di accettazione (continua)

ID	Descrizione	Stato
TA.2	<p>Verificare che il Tecnico possa salvare un nuovo <i>dizionario dati</i>₆ nel sistema:</p> <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari dati; 2. Avviare la procedura di inserimento di un nuovo dizionario; 3. Inserire il nome del dizionario; 4. Inserire la descrizione del dizionario; 5. Caricare il file <i>JSON</i>₆ associato al dizionario; 6. Confermare l'inserimento; 7. Visualizzare un messaggio di conferma una volta che il dizionario è stato salvato. 	S
TA.3	<p>Verificare che il Tecnico possa visualizzare la lista dei dizionari dati caricati nel sistema:</p> <ol style="list-style-type: none"> 1. Visualizzare la lista dei dizionari disponibili; 2. Visualizzare le caratteristiche di ciascun dizionario presente nella lista. 	S
TA.4	<p>Verificare che il Tecnico possa eliminare un dizionario dati dal sistema:</p> <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari caricati nel sistema; 3. Scegliere un dizionario tra quelli disponibili; 4. Richiedere l'eliminazione del dizionario; 5. Confermare la decisione; 6. Visualizzare un messaggio di conferma una volta che il dizionario è stato eliminato. 	S
TA.5	<p>Verificare che il Tecnico possa modificare il nome di un dizionario dati:</p> <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari caricati nel sistema; 3. Scegliere un dizionario tra quelli disponibili; 4. Modificare il nome del dizionario; 5. Confermare la modifica; 6. Visualizzare un messaggio di conferma una volta effettuata la modifica. 	S
Continua nella prossima pagina		

Tabella 3.5: Test di accettazione (continua)

ID	Descrizione	Stato
TA.6	<p>Verificare che il Tecnico possa modificare la descrizione di un dizionario dati:</p> <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari caricati nel sistema; 3. Scegliere un dizionario tra quelli disponibili; 4. Modificare la descrizione del dizionario; 5. Confermare la modifica; 6. Visualizzare un messaggio di conferma una volta effettuata la modifica. 	S
TA.7	<p>Verificare che il Tecnico possa modificare il file di configurazione di un dizionario dati:</p> <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari caricati nel sistema; 3. Scegliere un dizionario tra quelli disponibili; 4. Caricare un nuovo file <i>JSON_e</i>; 5. Confermare la sostituzione; 6. Visualizzare un messaggio di conferma una volta effettuata la modifica. 	S
TA.8	<p>Verificare che il Tecnico possa scaricare un <i>dizionario dati_e</i>:</p> <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari caricati nel sistema; 3. Scegliere un dizionario tra quelli disponibili; 4. Richiedere il download del file; 5. Visualizzare il file scaricato. 	S
TA.9	Verificare che il Tecnico possa effettuare correttamente il logout.	S
Continua nella prossima pagina		

Tabella 3.5: Test di accettazione (continua)

ID	Descrizione	Stato
TA.10	Verificare che l'Utente possa effettuare una ricerca per nome tra i dizionari dati: <ol style="list-style-type: none"> 1. Accedere alla chat o alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari disponibili; 3. Richiedere al sistema di effettuare una ricerca per nome; 4. Visualizzare i risultati della ricerca. 	S
TA.11	Verificare che il Tecnico possa effettuare una ricerca per descrizione tra i dizionari dati: <ol style="list-style-type: none"> 1. Accedere alla pagina di gestione dei dizionari; 2. Visualizzare la lista dei dizionari disponibili; 3. Richiedere al sistema di effettuare una ricerca per descrizione; 4. Visualizzare i risultati della ricerca. 	S
TA.12	Verificare che l'Utente possa selezionare un dizionario dati da utilizzare nell'applicazione.	S
TA.13	Verificare che l'Utente possa visualizzare un'anteprima del dizionario dati selezionato.	S
TA.14	Verificare che l'Utente possa ottenere un <i>prompt</i> _g in risposta a un'interrogazione in linguaggio naturale: <ol style="list-style-type: none"> 1. Accedere alla chat; 2. Inserire una richiesta in linguaggio naturale; 3. Inviare il messaggio; 4. Visualizzare il prompt generato dal sistema. 	S
TA.15	Verificare che l'Utente possa copiare il prompt generato: <ol style="list-style-type: none"> 1. Accedere alla chat; 2. Visualizzare il prompt generato; 3. Richiedere al sistema di copiare il contenuto del prompt; 4. Visualizzare un messaggio di conferma una volta che il prompt è stato copiato negli appunti. 	S
Continua nella prossima pagina		

Tabella 3.5: Test di accettazione (continua)

ID	Descrizione	Stato
TA.16	Verificare che l'Utente possa visualizzare il contenuto della chat: 1. Accedere alla chat; 2. Visualizzare la lista dei messaggi; 3. Visualizzare un singolo messaggio nella lista; 4. Visualizzare il mittente del messaggio; 5. Visualizzare il contenuto del messaggio.	S
TA.17	Verificare che l'Utente possa eliminare la cronologia della chat.	S
TA.18	Verificare che il Tecnico possa visualizzare il messaggio di <i>debug</i> _e elaborato durante la generazione del prompt.	S
TA.19	Verificare che il Tecnico possa scaricare un file di <i>log</i> _e .	S
TA.20	Verificare che il ChatBOT restituisca un messaggio esplicativo se la richiesta inserita dall'Utente non ha prodotto risultati rilevanti.	S

3.4.1 Tracciamento dei test di accettazione

Tabella 3.6: Tracciamento test di accettazione

ID	Caso d'uso
TA.1	UC1, UC1.1, UC1.2
TA.2	UC13, UC15, UC16, UC17
TA.3	UC9, UC9.1, UC10, UC10.1, UC10.2, UC10.3
TA.4	UC18
TA.5	UC29
TA.6	UC30
TA.7	UC20
TA.8	UC38
TA.9	UC12
TA.10	UC36
TA.11	UC37
TA.12	UC4
TA.13	UC14, UC14.1, UC14.2, UC14.3, UC14.3.1, UC14.3.1.1, UC14.3.1.2
TA.14	UC3, UC5, UC7, UC26
TA.15	UC8
TA.16	UC25, UC25.1, UC25.1.1, UC25.1.2
TA.17	UC27
TA.18	UC22
TA.19	UC23
TA.20	UC6

3.5 Checklist

Le checklist sono strumenti che affiancano il team nell'attività di ispezione del codice e della documentazione, al fine di accertarsi che questi siano conformi alle specifiche e alle linee guida (pratiche e stili di codifica, coerenza della documentazione). È un metodo di analisi statica mirato a individuare gli errori più ricorrenti che possono manifestarsi nel prodotto in esame.

3.5.1 Struttura della documentazione

Tabella 3.7: Checklist – Struttura della documentazione

Titolo	Descrizione
Riferimento a documenti soggetti al versionamento	Quando viene menzionato il contenuto di un documento soggetto al versionamento, il riferimento deve riportare, oltre al nome del documento, anche il numero di versione.
Riferimento a materiali online	Le risorse web, per loro stessa natura, sono mutevoli. Pertanto, il riferimento a materiali online deve riportare la data di ultimo accesso alla risorsa. Inoltre, il collegamento ipertestuale deve essere visibile direttamente come URL.
Didascalie ed etichette	Tutte le immagini e le tabelle devono essere corredate da una didascalia che ne descriva il contenuto, e da un'etichetta univoca che funga da riferimento globale.
Sezioni vuote o incomplete	Nessun documento deve contenere sezioni vuote o incomplete (ovvero sezioni il cui contenuto è descritto come "Todo").
Suddivisione indice	L'indice dei documenti deve essere suddiviso in tre sezioni: <ul style="list-style-type: none"> • Indice generale; • Elenco delle tabelle; • Elenco delle figure.
Occorrenze multiple di un termine nel <i>Glossario</i>	Quando un termine definito nel <i>Glossario</i> appare più volte all'interno di un documento, tutte le ricorrenze devono essere formattate in corsivo e marcate con una lettera _G in pedice (a meno di non compromettere la leggibilità).
Occorrenze multiple di un termine nel <i>Glossario</i> (verbali esterni)	Quando un termine definito nel <i>Glossario</i> appare più volte all'interno di un verbale esterno, solamente la prima ricorrenza deve essere formattata.
Introduzione <i>Glossario</i> (verbali esterni)	L'introduzione del <i>Glossario</i> nei verbali esterni deve essere diversa rispetto a quella degli altri documenti.
Continua nella prossima pagina	

Tabella 3.7: Checklist - Struttura della documentazione
(continua)

Titolo	Descrizione
Punteggiatura elenchi puntati o numerati	La frase di introduzione di un elenco puntato o numerato deve terminare con i due punti. Le voci di un elenco, invece, devono finire con un punto se rappresentano la conclusione dell'elenco o sotto-elenco in questione, altrimenti con un punto e virgola.
Formato delle date	Tutte le date non incluse in un paragrafo discorsivo devono apparire nella forma "AAAA-MM-GG".
Indice di leggibilità	Le modifiche ai documenti devono rispettare la soglia di tollerabilità stabilita per l'Indice Gulpease.
Distribuzione verbali esterni	Nella distribuzione dei verbali esterni deve essere menzionata, oltre al gruppo fornitore e ai Committenti, anche la Proponente.
Ordinamento registro modifiche per data	Nel changelog, le modifiche devono essere ordinate dalla più recente alla più vecchia.
Ordinamento task per ID	I task devono essere disposti in ordine crescente sulla base del loro ID.
Menzione di un soggetto	Quando si menziona una persona, la formula da utilizzare è la seguente: "Nome Cognome". Per mantenere coerenza all'interno dei documenti e sfruttare i comandi $LaTeX$, il team ha adottato questa formula anche negli elenchi e nelle tabelle. Pertanto, i nomi non vengono disposti in ordine alfabetico, ma seguono l'ordinamento definito nel template globale.

3.5.2 Errori formali

Tabella 3.8: Checklist - Errori formali

Titolo	Descrizione
Nomi dei ruoli di progetto	I nomi dei ruoli di progetto devono avere la lettera iniziale minuscola.
Continua nella prossima pagina	

Tabella 3.8: Checklist – Errori formali (continua)

Titolo	Descrizione
Proponente	Il termine Proponente deve iniziare con la lettera maiuscola e, in via preferenziale, essere declinato al femminile. Tuttavia, l'uso maschile di Proponente è ritenuto corretto.
Cliente e Committente	I termini Cliente e Committente richiedono l'iniziale maiuscola solamente quando si riferiscono a un attore specifico e non a un ruolo o entità astratta.
Repository	Il termine repository deve essere declinato al maschile.
IA e AI	Possono essere utilizzati entrambi gli acronimi (intelligenza artificiale o, come indicato nel capitolato, artificial intelligence).
Back-end/backend e front-end/frontend	Si possono utilizzare entrambe le forme, con o senza trattino.
Sintassi e ortografia	Il testo deve essere privo di errori di sintassi e ortografia.
<i>typo</i> _e	È essenziale limitare gli errori tipografici e le sviste, specialmente nella scrittura del codice.
Linguaggio	I documenti devono essere redatti in modo impersonale dal punto di vista della forma verbale. Inoltre, è opportuno adottare un linguaggio il più formale possibile, soprattutto nella stesura dei verbali esterni.
Versioni estese di abbreviazioni	Le versioni estese delle sigle devono rispettare la forma delle abbreviazioni (es.: AdR diventa Analisi dei requisiti, WoW diventa Way of Working).
Soggetto della frase	Il soggetto di un discorso deve sempre essere evidenziato nella sua introduzione.
D eufonica	La d eufonica deve essere inserita solo quando le due vocali sono uguali.
PoC	Nonostante la traduzione di Proof of Concept (PoC) sia "verifica teorica" o "prova di fattibilità", il termine PoC deve essere declinato al maschile.
Open-source e open source	Entrambe le forme sono accettate, con o senza trattino.
Continua nella prossima pagina	

Tabella 3.8: Checklist – Errori formali (continua)

Titolo	Descrizione
Consistenza nell'uso delle lettere maiuscole nei titoli	Nei titoli delle sezioni o dei paragrafi dei documenti, la lettera iniziale maiuscola è riservata solo alla prima parola, salvo disposizioni contrarie nelle <i>Norme di Progetto</i> .
Declinazione di termini provenienti dalla lingua inglese	I termini inglesi inseriti all'interno di un documento italiano non vanno declinati (salvo rare eccezioni, ad esempio embeddings), in quanto la lingua italiana non prevede la formazione del plurale tramite l'aggiunta della desinenza -s o -es.
Componente	Il termine componente può essere declinato sia al maschile che al femminile.
ChatBOT	Il termine ChatBOT deve seguire la convenzione utilizzata per la scrittura di ChatGPT ("Chat" + "GPT").
Web-based e web based	Entrambe le forme sono corrette, con o senza trattino.

3.5.3 Analisi dei Requisiti

Tabella 3.9: Checklist – Analisi dei Requisiti

Titolo	Descrizione
Correlazione casi d'uso - requisiti	Ciascun caso d'uso dovrebbe essere associato a uno o più requisiti.
Ordinamento requisiti	I requisiti devono essere ordinati secondo la stessa disposizione dei casi d'uso.
Diagrammi dei casi d'uso	Le inclusioni, estensioni e relazioni di generalizzazione dovrebbero essere rappresentate in un unico diagramma <i>UML_G</i> . Tuttavia, per motivi di spazio, possono essere riportate anche in diagrammi separati.
Coerenza diagramma-descrizione UC	Il diagramma UML e la descrizione dei casi d'uso devono essere consistenti.
Continua nella prossima pagina	

Tabella 3.9: Checklist – Analisi dei Requisiti (continua)

Titolo	Descrizione
Completezza descrizione UC	La descrizione dei casi d’uso deve essere esaustiva, integrando le informazioni già riportate nel diagramma UML.
Distinzione tra requisiti funzionali e non funzionali	La separazione tra i requisiti funzionali e non funzionali deve essere chiara.
Tracciamento dei requisiti	Ogni requisito deve essere ricavato da almeno una fonte. Il tracciamento dei requisiti (composto dalle coppie requisito-fonti) deve essere privo di errori.

3.5.4 Codifica

Tabella 3.10: Checklist – Codifica

Titolo	Descrizione
Nomi esplicativi	I nomi di classi, metodi, attributi e variabili devono essere “parlanti”, in quanto rappresentano la prima forma di documentazione del codice.
Regole di nomenclatura	Le regole specificate nelle <i>Norme di Progetto</i> devono essere rispettate. Tali regole possono includere: <ul style="list-style-type: none"> • Uso di CamelCase (es.: firstName) o snake_case (es.: first_name); • Prefissi o suffissi per tipo di dato; • Convenzioni per le costanti.
Numerosità dei commenti	Porzioni di codice o metodi rilevanti dovrebbero essere preceduti da un commento. Per contro, è opportuno evitare commenti superflui che non migliorano la leggibilità.
Commenti significativi	I commenti devono essere significativi, ossia devono fornire in modo diretto informazioni utili sul funzionamento del codice.

4 Cruscotto di valutazione della qualità

4.1 M.PC.1 – Percentuale di metriche soddisfatte

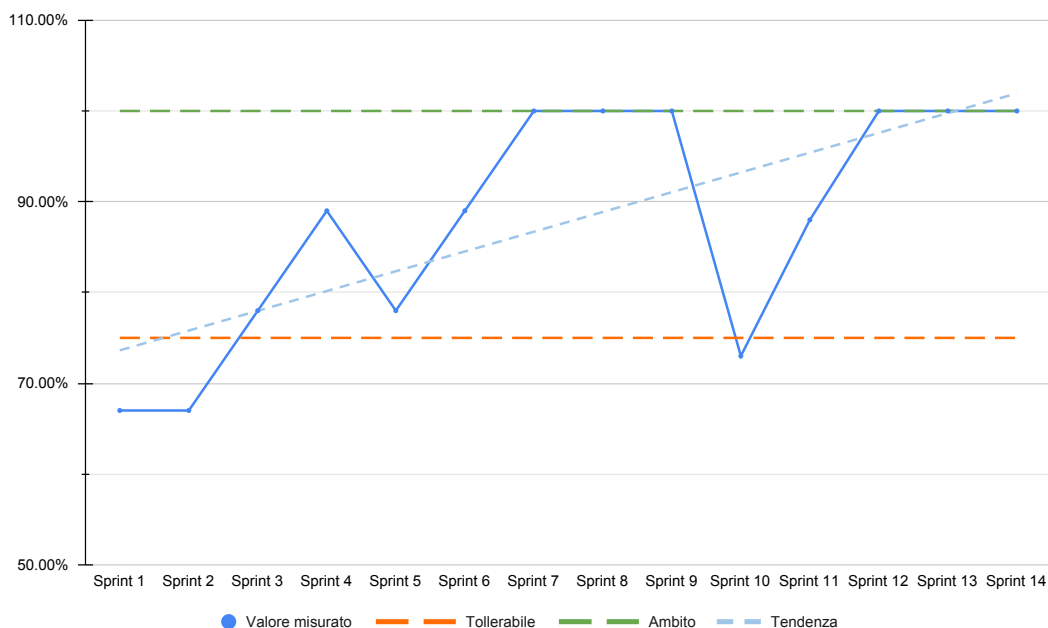


Figura 4.1: M.PC.1 – Percentuale di metriche soddisfatte

Il grafico sottolinea come negli *sprint*₀ iniziali il team non abbia raggiunto la soglia di tollerabilità stabilita. Il mancato raggiungimento degli obiettivi era dovuto all'inesperienza del gruppo e alla difficoltà di adattamento ai processi di lavoro e alle pratiche dell'ingegneria del software. Tuttavia, nel corso degli *sprint*₀ successivi, il gruppo ha notato dei progressi, specie nell'ottemperanza alle metriche di variazione del piano e del budget. Questo è dovuto all'introduzione di feedback migliorativi, alla maggiore competenza e collaborazione all'interno del team, e all'applicazione più puntuale del ciclo di PDCA. Inoltre, il gruppo ha aggiornato e approfondito il Way of Working nelle *Norme di Progetto*, incrementando la qualità dei processi. Il grafico illustra un miglioramento costante nel tempo, fino a raggiungere il valore ambito nel settimo sprint. Già dal terzo sprint, però, il valore misurato era superiore alla soglia considerata tollerabile.

Con l'introduzione del monitoraggio delle metriche di prodotto, il team è riuscito a mantenere un livello di qualità costante. L'unica eccezione si è verificata durante il decimo *sprint*₀, in cui la percentuale di metriche soddisfatte è scesa al di sotto della soglia accettabile. Questo calo è dovuto all'inesperienza del gruppo nelle fasi di progettazione e testing, fondamentali per il raggiungimento degli standard previsti.

A partire dallo sprint 12, tuttavia, la percentuale di metriche soddisfatte è tornata al 100%.

4.2 M.PC.5 – EAC (Estimated at Completion)

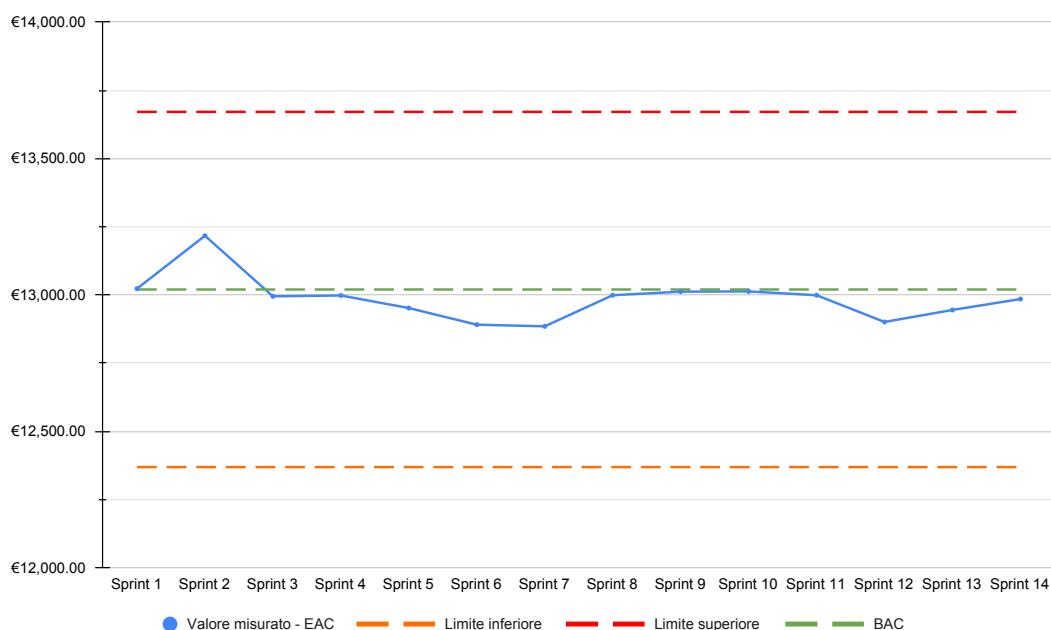


Figura 4.2: M.PC.5 – EAC (Estimated at Completion)

Nella fase iniziale del progetto, l'EAC (costo sostenuto + stima costo ancora da sostenere) è rimasto in linea con il BAC (valore iniziale previsto), con l'unica eccezione rappresentata dal secondo *sprint*. Durante la pianificazione del secondo sprint, infatti, il gruppo ha riscontrato delle carenze strutturali nella documentazione. Pertanto, gli sforzi del team si sono concentrati sulla rettifica dei documenti prodotti fino a quel momento. In particolare, la riorganizzazione dell'*Analisi dei Requisiti*, del *Piano di Progetto* (con l'aggiunta della gestione dei rischi e del preventivo "a finire") e dei verbali (con l'aggiunta della tabella "Todo"), hanno richiesto un impegno maggiore del previsto. Inoltre, lo studio dei nuovi strumenti e tecnologie (tra cui *Jira*, *YAML* e *txtai*) ha rallentato l'avanzamento dei lavori, mantenendo però i costi elevati. Nonostante il budget stimato per la realizzazione del progetto abbia superato il BAC, il team ha comunque rispettato il range di tollerabilità stabilito. A partire dal terzo sprint, invece, la stima del budget è risultata inferiore rispetto al BAC, avvicinandosi a quest'ultimo nell'ottavo sprint (coincidente con la sessione di esami). L'andamento del grafico denota come le azioni preventive e correttive impiegate dal team abbiano avuto esito positivo, garantendo il rispetto del budget nonostante il cambio tecnologico avvenuto nel quinto sprint.

Nell'arco della *PB*, la pianificazione accurata e l'elevato impegno lavorativo hanno permesso al gruppo di rispettare il budget previsto senza incorrere in ritardi.

4.3 M.PC.6 – Variazione del budget tra preventivo e consuntivo



Figura 4.3: M.PC.6 – Variazione del budget tra preventivo e consuntivo

Nel primo *sprint*₆, il gruppo ha sovrastimato il carico di lavoro necessario per svolgere le attività assegnate al ruolo di analista; pertanto, i costi effettivi sono risultati inferiori rispetto a quanto preventivato. La medesima situazione si è verificata nel terzo sprint, ma con risultato opposto. La progettazione e l'implementazione delle funzionalità del *PoC*₆ (correlate alla libreria *txtai*₆) hanno comportato un aumento delle ore riservate ai ruoli di programmatore e progettista. Di conseguenza, il team ha superato i costi stimati. Nelle iterazioni successive, invece, il gruppo ha lavorato rispettando i costi allocati in fase di preventivo. A differenza di altre metriche, la variazione del budget non ha mai sfiorato il range di tollerabilità, avvicinandosi al valore ambito in concomitanza del sesto sprint. Per questo motivo, il team ha riformulato il valore tollerabile, abbassandolo da $\pm 15\%$ a $\pm 10\%$. Nella maggior parte degli sprint, la variazione è stata maggiore di 0; ciò significa che il gruppo ha speso il proprio budget con minor velocità di quanto pianificato. L'obiettivo del team è di diminuire lo scostamento, sia in positivo che in negativo, migliorando la pianificazione dei task, la distribuzione dei ruoli e la stima oraria delle attività.

Nel corso della *PB*₆, il gruppo è riuscito a raggiungere gli obiettivi prefissati, riducendo lo scostamento tra il budget stimato e quello effettivamente speso. Questo risultato dimostra come l'esperienza maturata dal team e le misure adottate abbiano contribuito a un miglioramento continuo dei processi. L'unica eccezione è stata lo *sprint*₆ 13, durante il quale il gruppo si è concentrato sulla preparazione per la revisione *PB*, rimandando alcune attività già pianificate.

4.4 M.PC.7 - Variazione del piano tra preventivo e consuntivo

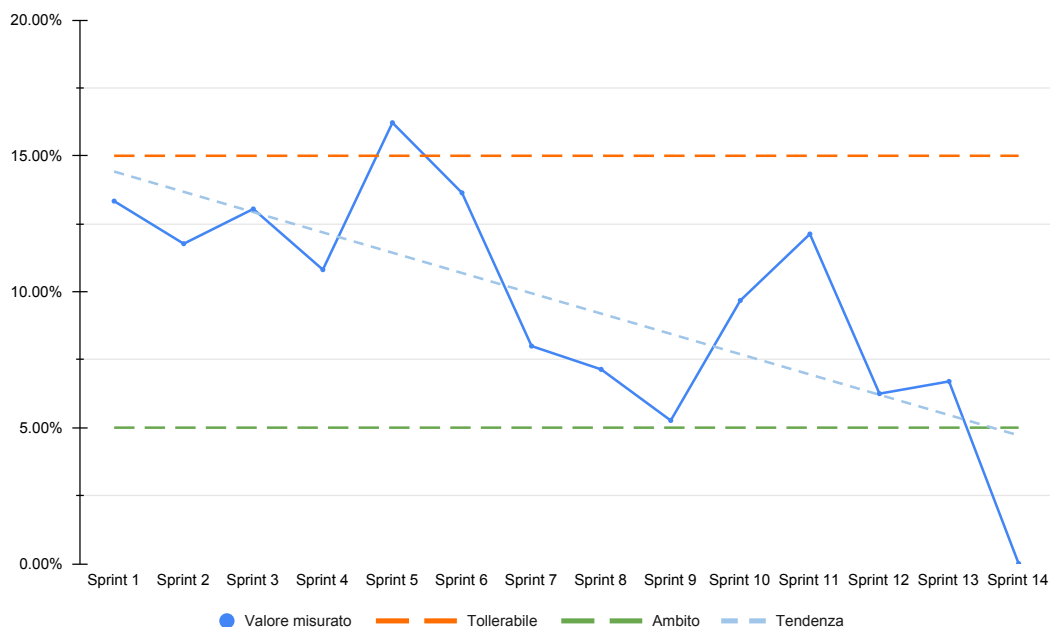


Figura 4.4: M.PC.7 - Variazione del piano tra preventivo e consuntivo

Il gruppo ha mantenuto un sano equilibrio nel rapporto tra le attività pianificate a inizio *sprint*_e e quelle completate. Il risultato è indice di una pianificazione iniziale discretamente accurata, anche se non ancora ottimale. Negli sprint iniziali (della durata di due settimane), il team ha faticato ad avvicinarsi al valore ambito; questo per via di un carico di lavoro eccessivo assegnato ad alcuni membri e ruoli di progetto. L'ampia durata degli sprint ha comportato una pianificazione non prevedente e troppo ambiziosa. Nonostante ciò, la discrepanza tra i task pianificati e quelli completati è rimasta entro i parametri di accettabilità. Il picco in corrispondenza del quinto *sprint*_e è dovuto alla sua durata minore, per cui la pianificazione iniziale non è stata bilanciata adeguatamente. A partire dalla misura successiva si è tuttavia ricalibrato il carico di lavoro, rientrando nel range di tollerabilità. Superato il debito tecnico dovuto al cambio di tecnologie, il team è riuscito ad avvicinarsi al valore ambito.

Nell'undicesimo *sprint*_e, il valore misurato si è allontanato dalla soglia desiderata a causa del tempo richiesto per la configurazione dei test. Nel corso dello sprint, infatti, la maggior parte delle attività di testing *back-end*_e sono state posticipate. A partire dal dodicesimo sprint, la differenza tra i task pianificati e quelli completati si è nuovamente ridotta, raggiungendo e superando il valore desiderato nello sprint 14.

4.5 M.PC.8 - Efficienza temporale

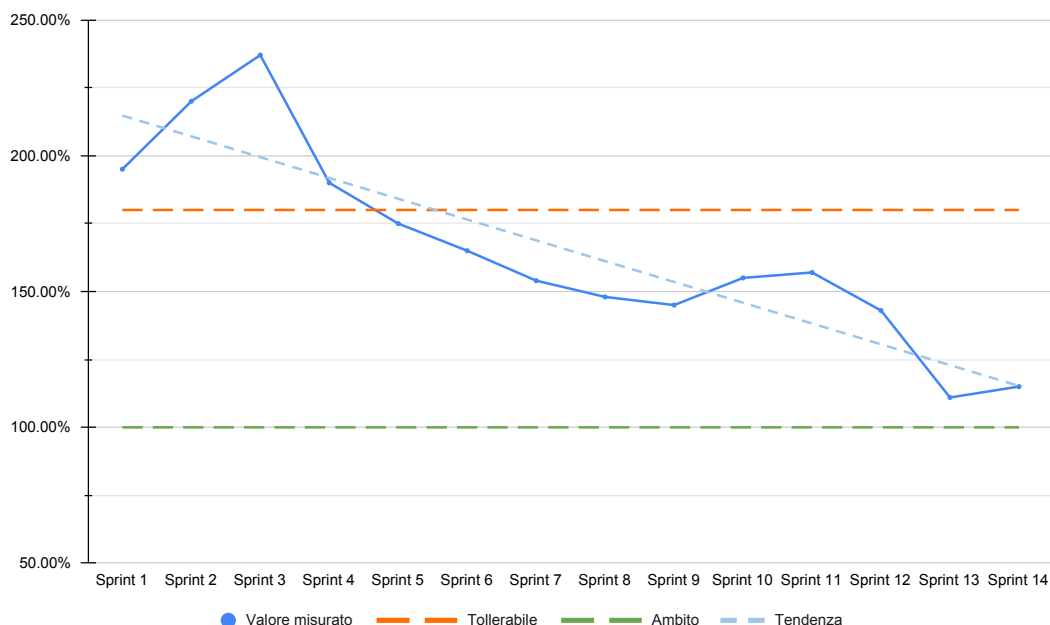


Figura 4.5: M.PC.8 - Efficienza temporale

Il grafico mostra una performance iniziale al di sotto delle aspettative, seguita da un miglioramento significativo nel corso del tempo. Nella prima fase del progetto, il gruppo ha speso un numero di ore che si è tradotto solo in minima parte in ore produttive, indicando possibili inefficienze o adattamenti necessari per la formazione e ricerca degli strumenti. Tuttavia, con il progredire degli *sprint_g*, si osserva un incremento costante dell'efficienza temporale. I fattori di questo miglioramento sono l'inclusione di pratiche di ottimizzazione dei processi, l'introduzione di nuovi strumenti e tecnologie, e un aumento della familiarità e della coesione del team. L'aumento dell'efficienza ha consentito al team di ridurre il valore tollerabile dal 200% al 180%.

Dopo un inizio con prestazioni inferiori alle attese, il grafico dell'efficienza temporale testimonia un percorso di miglioramento, che ha portato il team a conseguire una produttività e un ritmo apprezzabili. Questo evidenzia non soltanto una crescita in termini di rendimento, ma dimostra soprattutto l'adattabilità e la capacità di apprendimento del gruppo nel massimizzare le risorse a disposizione. Durante la *PB_g*, il team ha mantenuto un'efficienza costante, contribuendo a stabilizzare la traiettoria del grafico. A partire dallo *sprint_g* 12, il gruppo ha migliorato ulteriormente l'efficienza temporale, avvicinandosi al valore desiderato.

4.6 M.PC.9 – Frequenza di merge delle pull request

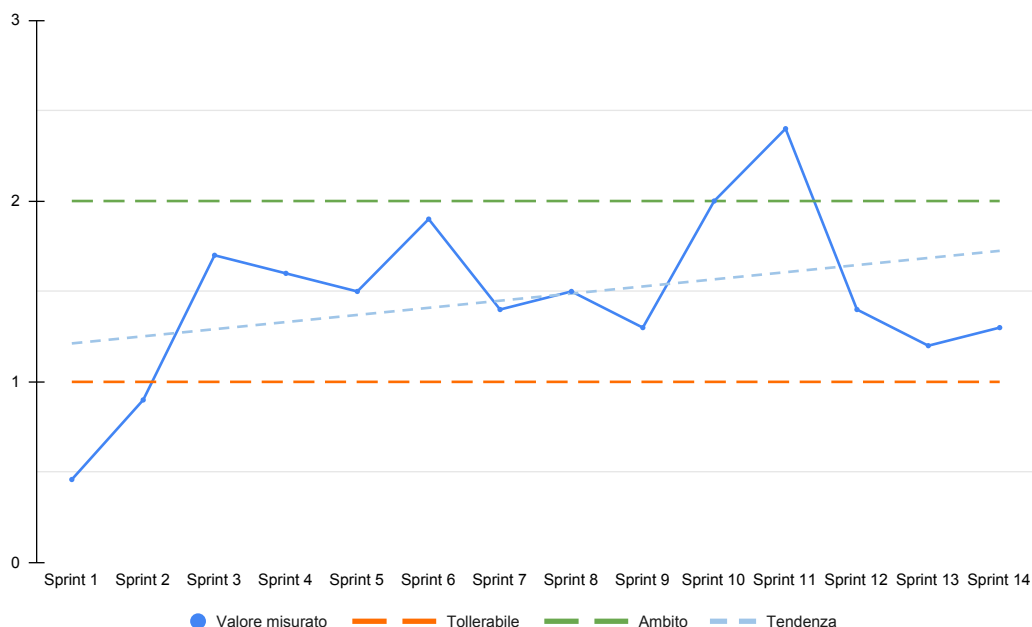


Figura 4.6: M.PC.9 – Frequenza di merge delle pull request

Nei primi due *sprint_G*, la frequenza di merge delle pull request è stata inferiore alle aspettative; questo è dovuto al fatto che, nelle fasi iniziali del progetto, gli sforzi del gruppo si sono concentrati sulla produzione di documenti. Data l'inesperienza del team nella stesura della documentazione, le pull request sono state aperte con discontinuità. Inoltre, la portata delle modifiche ha rallentato il processo di verifica e approvazione. Pertanto, la frequenza di merge non ha raggiunto il valore tollerabile (1 volta al giorno). Il team ha quindi deciso di ridurre il valore ambito, da 3 volte al giorno a 2. Dal terzo sprint, vista la necessità di aggiornare specifiche sezioni dei documenti, il gruppo ha ritenuto opportuno integrare le modifiche con maggior frequenza. È stata introdotta la pratica di *continuous integration_G*, migliorando il processo di allineamento delle modifiche e consentendo verifiche rapide e frequenti. Un fattore che ha contribuito a incrementare la frequenza di merge è stato lo sviluppo del *PoC_G*, le cui funzionalità sono state suddivise in task di dimensioni ridotte, al fine di promuovere l'integrazione continua. Grazie all'applicazione di questa contromisura, il team ha mantenuto un flusso di lavoro regolare. Come testimonia il grafico, i valori misurati a partire dal terzo sprint rientrano nel range di tollerabilità stabilito. In concomitanza del sesto sprint, la frequenza di merge delle pull request si è avvicinata al valore ambito; considerando il cambio di tecnologie avvenuto nell'iterazione precedente, questo risultato dimostra l'efficacia delle strategie adottate dal gruppo.

Nella prima fase della *PB_G*, la frequenza di merge delle pull request ha raggiunto la soglia ambito. Questo risultato è in linea con la decisione del team di applicare con

maggior rigore la pratica di *integrazione continua*₆. Nel dodicesimo sprint, invece, il valore misurato è rimasto al di sotto della soglia desiderata, poiché l'integrazione delle modifiche (tramite il meccanismo delle pull request) ha richiesto più tempo e più revisioni del previsto. Nonostante ciò, la frequenza di merge è rimasta entro i limiti di tollerabilità.

4.7 M.PC.10 – Indice di stabilità dei requisiti

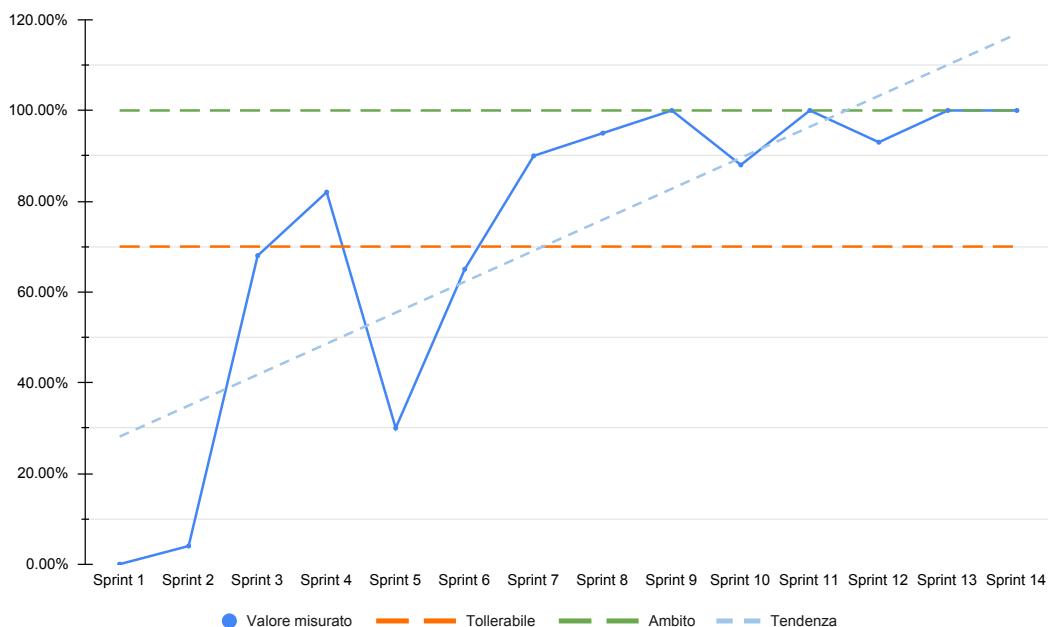


Figura 4.7: M.PC.10 – Indice di stabilità dei requisiti

Il grafico evidenzia un'alta instabilità dei requisiti durante i primi due *sprint*, a causa dell'inesperienza del team nella definizione dei *casì d'uso* e nell'identificazione dei requisiti funzionali e non funzionali. Inoltre, sono state apportate modifiche sostanziali ai requisiti dopo l'incontro con la *Proponente*. Dal terzo *sprint*, invece, si nota un incremento nella stabilità, in quanto le modifiche hanno avuto un impatto minore. Nel quinto sprint, a seguito di un incontro con il Professor Cardin, il team ha attuato una rivisitazione completa e una ristrutturazione del documento di *Analisi dei Requisiti*. Sono stati inoltre aggiunti nuovi casi d'uso e requisiti, poiché la stretta collaborazione tra i membri del team ha contribuito a chiarire le funzionalità del prodotto. Pertanto, in corrispondenza del quinto sprint, il valore misurato risulta essere nuovamente inferiore alle aspettative. A partire dal settimo sprint, il gruppo ha riportato la situazione sotto controllo, raggiungendo un livello di profondità e stabilità dei requisiti soddisfacente.

Durante il decimo *sprint*, il documento di *Analisi dei Requisiti* è stato aggiornato in seguito alla revisione *RTB*. In particolare, è stato eliminato un requisito ed è stata migliorata la suddivisione tra requisiti funzionali, di vincolo e di qualità. Nel dodicesimo sprint, il team ha effettuato ulteriori aggiornamenti ai requisiti basandosi sulle osservazioni dei programmatori. Questo ha comportato la revisione di alcuni requisiti, tra cui: la copia del messaggio di *debug* e il controllo degli errori nella descrizione del *dizionario dati*. Nonostante queste modifiche, il team è riuscito a mantenere l'indice di stabilità dei requisiti vicino al valore desiderato.

4.8 M.PC.11 – Rischi inattesi

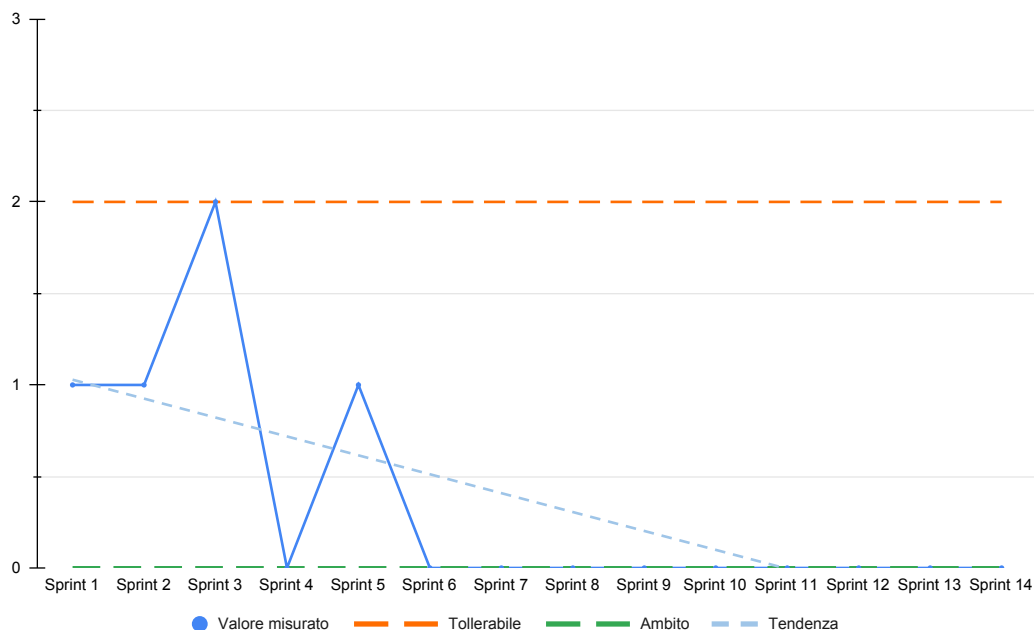


Figura 4.8: M.PC.11 – Rischi inattesi

Il grafico evidenzia l'inesperienza iniziale del team nell'individuare i rischi che possono emergere durante lo svolgimento di un progetto software. Nei primi tre *sprint*, infatti, il gruppo ha dovuto affrontare almeno un rischio inatteso. Ciononostante, il numero di rischi imprevisti è rimasto entro i limiti della soglia tollerabile. A partire dal quarto sprint, i rischi che si sono verificati erano già stati analizzati e documentati nel *Piano di Progetto*. Attraverso un'analisi più consapevole, una collaborazione stretta tra i membri e una comunicazione trasparente, il team ha mantenuto il numero di eventi imprevisti stabile e prossimo al valore ambito. L'unica eccezione è stata il quinto sprint, durante il quale è emerso un rischio inatteso legato al cambio di tecnologie. Nonostante il gruppo avesse previsto una possibile transizione e avesse testato diversi *framework* alternativi, l'entità del lavoro risultante ha superato le risorse disponibili, prolungando le scadenze prefissate. Per migliorare la gestione del progetto, il gruppo ha convenuto di discutere e monitorare i rischi durante le riunioni interne, fornendo al responsabile una base solida per la stesura del *Piano di Progetto*.

Nel corso della *PB*, il team ha mantenuto una gestione stabile dei rischi. Nonostante l'elevato numero di attività da svolgere, inclusa la configurazione dei test automatici, il gruppo non ha rilevato la comparsa di rischi inattesi.

4.9 M.PC.12 – Efficacia delle contromisure nei rischi

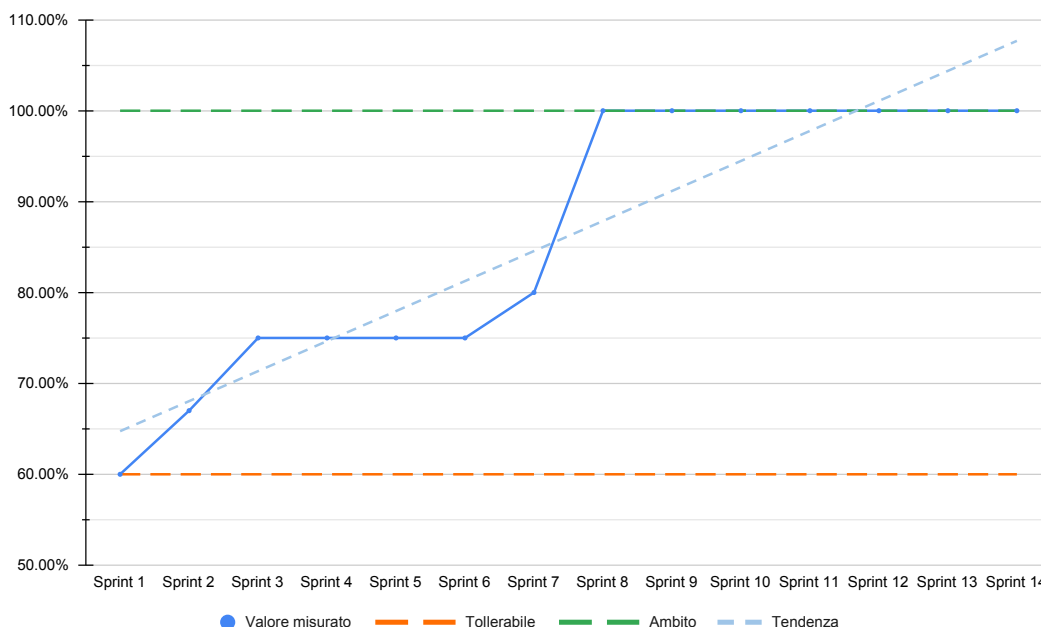


Figura 4.9: M.PC.12 – Efficacia delle contromisure nei rischi

L'andamento dell'efficacia delle contromisure descrive un percorso di crescita da una fase iniziale di adattamento e apprendimento a una fase successiva di miglioramento continuo. Nei primi *sprint*, le contromisure applicate non risultavano sufficienti per garantire una puntuale e completa gestione dei rischi, evidenziando l'insperienza dell'analisi preliminare. A partire dal terzo sprint, il team ha sperimentato nuove strategie e procedure mirate ad affrontare e mitigare le difficoltà operative. Per ciascun rischio identificato, sono state introdotte almeno due contromisure, in modo tale da disporre dei cosiddetti fallback (opzioni di emergenza da adottare quando la soluzione primaria fallisce). Con l'accumulo di esperienza e l'introduzione di contromisure più specifiche, l'andamento dell'efficacia delle contromisure ha mostrato segnali di miglioramento. Un ulteriore problema rilevato dal team riguardava l'applicazione tardiva delle contromisure. Pertanto, sono state adottate nuove strategie di rilevamento, al fine di identificare e gestire i rischi con maggior tempestività. Queste strategie includono best practices e tecnologie applicate grazie ai feedback provenienti dall'esperienza pratica e dall'osservazione diretta. L'andamento del grafico testimonia la validità delle soluzioni applicate dal gruppo. Dopo un periodo iniziale di adattamento, infatti, l'impatto dei rischi è diminuito progressivamente.

4.10 M.PD.4 - Indice Gulpease

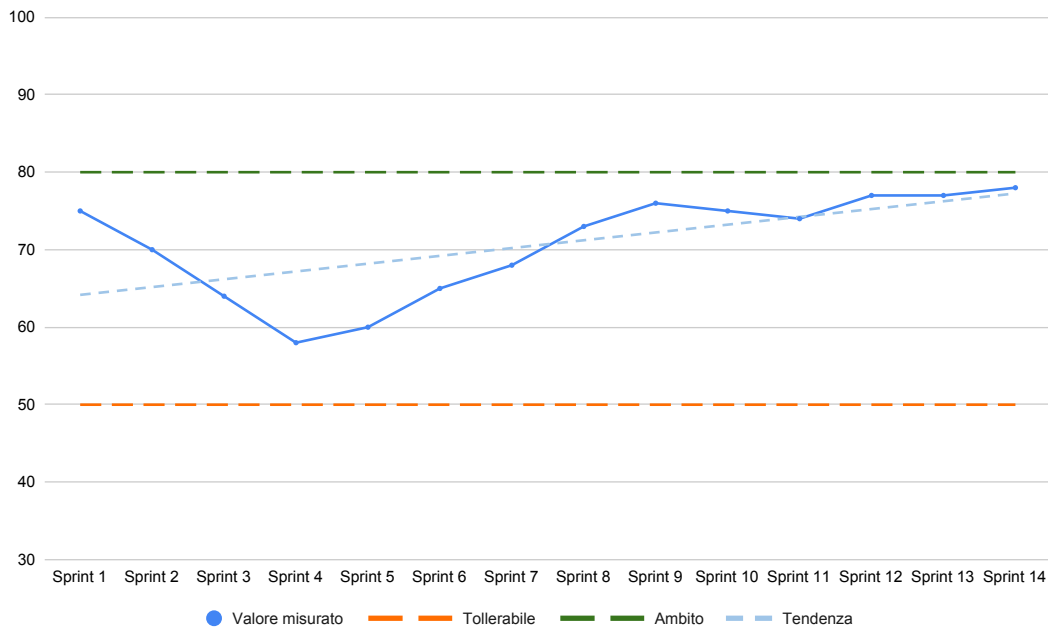


Figura 4.10: M.PD.4 - Indice Gulpease

Il grafico riporta la media degli Indici Gulpease di tutti i documenti, sia interni che esterni. Il valore medio oscilla tra 55 e 78; ciò significa che i documenti sono comprensibili anche per chi possiede una licenza media. I valori più bassi sono stati rilevati nella fase centrale della *RTB_G*, poiché il team ha apportato modifiche sostanziali alla documentazione senza applicare un controllo rigoroso sulla leggibilità. Tuttavia, i valori misurati rientravano nella soglia di tollerabilità. Per quanto concerne i singoli documenti, l'*Analisi dei Requisiti* ha evidenziato un indice di leggibilità inferiore rispetto agli altri nella *RTB*, in quanto conteneva frasi lunghe, specialmente nelle tabelle dei requisiti.

Nel corso della *PB_G*, il gruppo ha rielaborato i segmenti più prolissi del documento di *Analisi dei Requisiti*, ottenendo un Indice Gulpease di 75. I documenti introdotti nella *PB*, ovvero il *Manuale Utente* e la *Specifiche Tecnica*, hanno un punteggio superiore a 90, poiché il team ha applicato fin da subito una struttura e uno stile orientati alla leggibilità. L'Indice Gulpease complessivo è rimasto costante durante la *PB*, avvicinandosi al valore ambito nello *sprint_G* 14. Il *Glossario* ha un indice di leggibilità di 50; tale punteggio è dovuto alla presenza di termini tecnici che richiedono una definizione approfondita. Di seguito è riportato l'Indice Gulpease dei singoli documenti (per i verbali viene menzionato il valore medio).

Tabella Indice Gulpease – Ultimo aggiornamento: 2024-09-19

Documento	Indice Gulpease
<i>Piano di Progetto v2.0.0</i>	69
<i>Piano di Qualifica v2.0.0</i>	76
<i>Norme di Progetto v2.0.0</i>	74
<i>Analisi dei Requisiti v2.0.0</i>	75
<i>Glossario v2.0.0</i>	50
<i>Specifica Tecnica v1.0.0</i>	91
<i>Manuale Utente v1.0.0</i>	93
<i>Verballi interni</i>	81
<i>Verballi esterni</i>	70

Tabella 4.1: Tabella Indice Gulpease

4.11 M.PD.1 – Percentuale di requisiti obbligatori soddisfatti

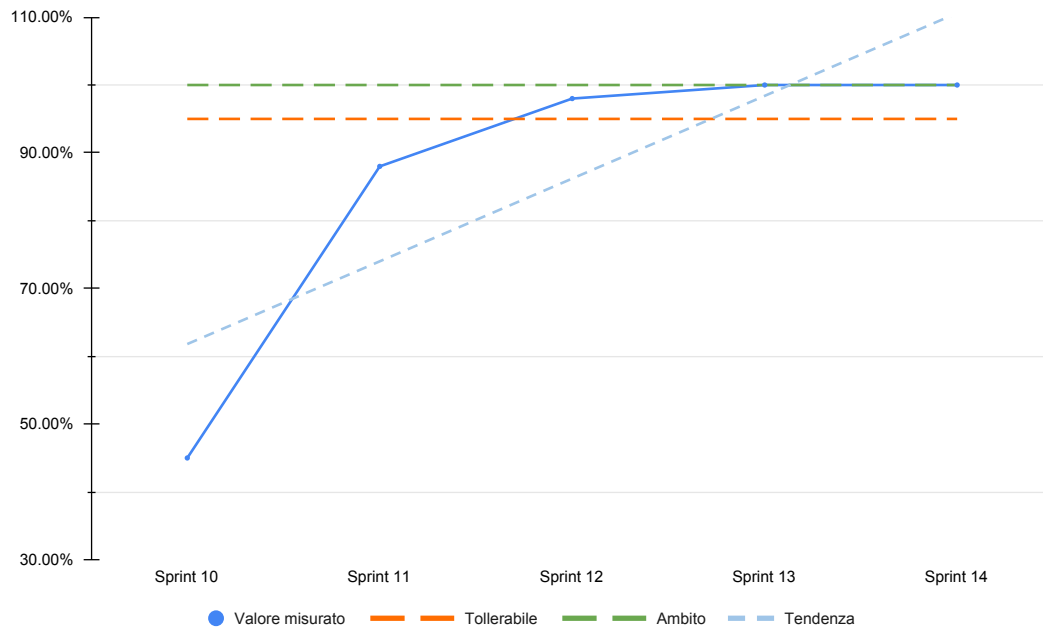


Figura 4.11: M.PD.1 – Percentuale di requisiti obbligatori soddisfatti

Nei primi due sprint della PB_{ϵ} , il team ha raggiunto un buon livello di copertura dei requisiti obbligatori, pur rimanendo al di sotto della soglia tollerabile. Nonostante i rapidi progressi nello sviluppo del $front-end_{\epsilon}$, l'assenza di un'architettura $back-end_{\epsilon}$ ben definita non consentiva il completo soddisfacimento dei requisiti. Dopo l'implementazione del modello architetturale e la configurazione dei test, il gruppo è riuscito a ottenere una copertura adeguata dei requisiti obbligatori. La piena copertura dei requisiti è stata infine raggiunta durante lo $sprint_{\epsilon}$ 13.

4.12 M.PD.2 – Percentuale di requisiti desiderabili soddisfatti

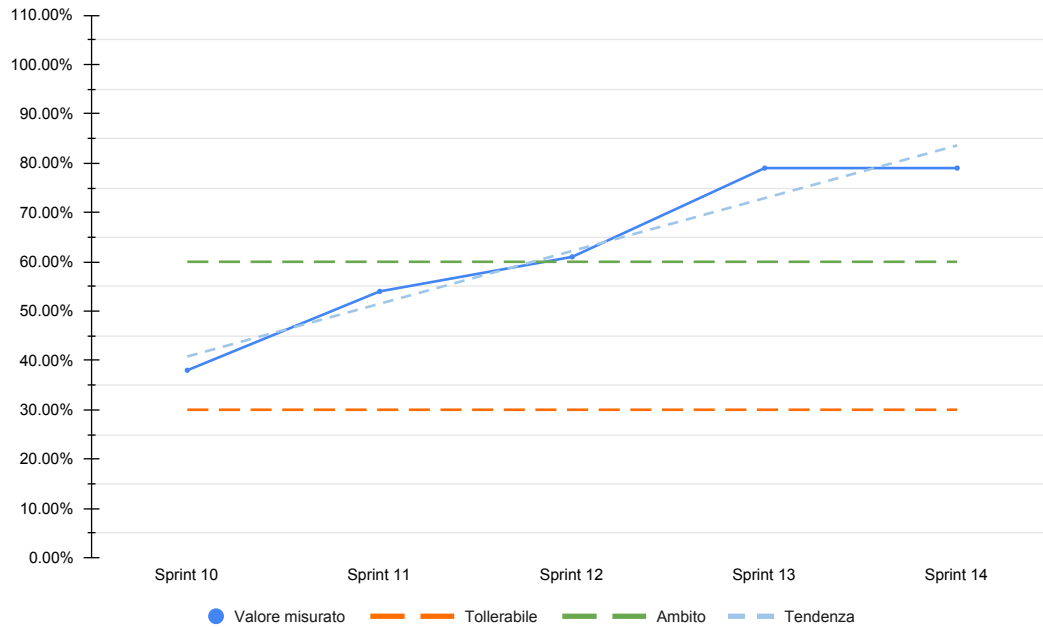


Figura 4.12: M.PD.2 – Percentuale di requisiti desiderabili soddisfatti

Nell'arco della PB_e , la copertura dei requisiti desiderabili è rimasta costantemente entro i limiti di tollerabilità. Ciò è stato possibile grazie al fatto che alcuni di questi requisiti, come l'aggiunta di una funzionalità di *debug_e*, erano già stati soddisfatti durante la RTB_e . Questo ha permesso al team di migliorare progressivamente la percentuale di copertura, fino a superare il valore ambito. I requisiti non soddisfatti riguardano principalmente aspetti di accessibilità, come il contrasto dei colori e la visualizzazione di informazioni aggiuntive riguardanti i *dizionari dati_e*.

4.13 M.PD.3 – Percentuale di requisiti opzionali soddisfatti

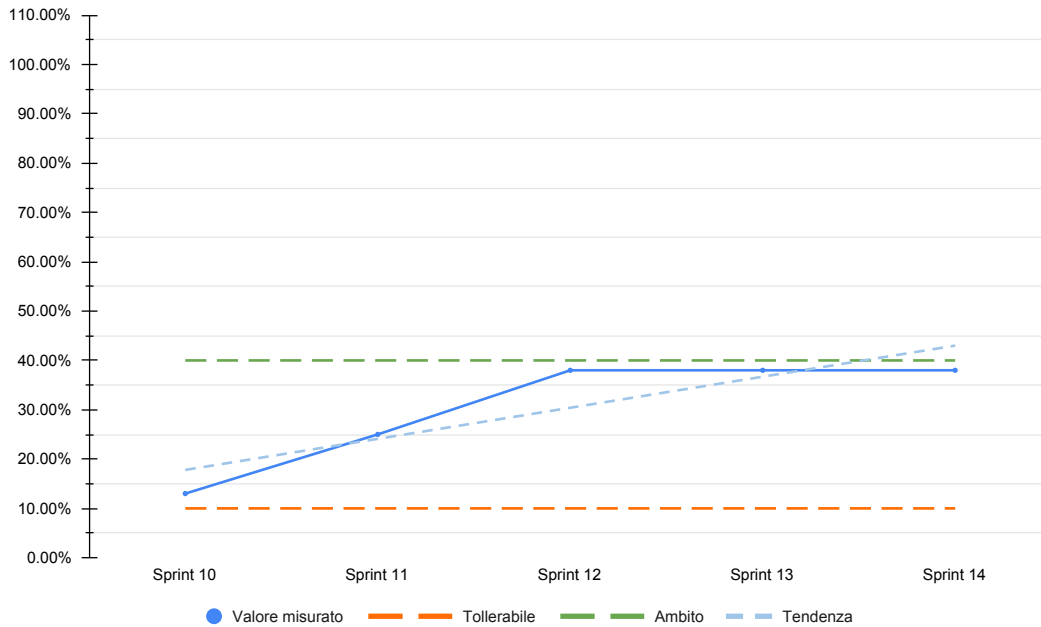


Figura 4.13: M.PD.3 – Percentuale di requisiti opzionali soddisfatti

Il team ha raggiunto una copertura dei requisiti opzionali inferiore al 50%. Questo risultato è dovuto al fatto che alcuni di questi requisiti richiedevano un abbonamento a pagamento per accedere alle *API* dei modelli di intelligenza artificiale. Inoltre, il gruppo ha scelto di concentrarsi su funzionalità per le quali la *Proponente* ha manifestato interesse durante le riunioni, come il cambio lingua e il *debug* del *prompt*. Nonostante ciò, la copertura dei requisiti opzionali è rimasta entro un intervallo considerato accettabile.

4.14 M.PD.5 - Completezza descrittiva

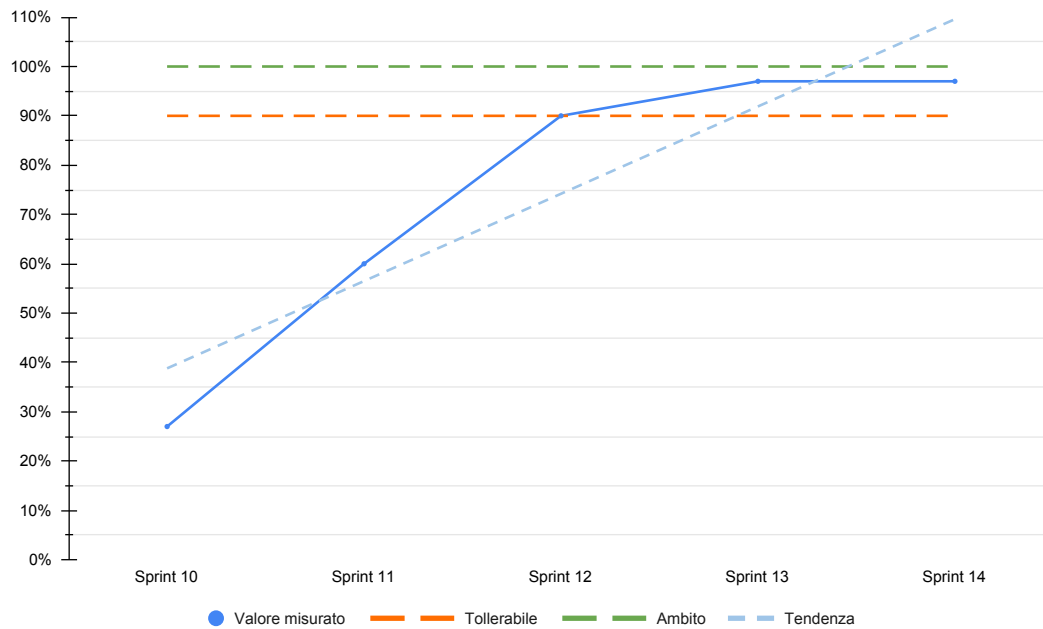


Figura 4.14: M.PD.5 - Completezza descrittiva

Nei primi due sprint della PB_e , le funzioni sviluppate non sono state documentate adeguatamente, poiché il team era focalizzato sulle attività di progettazione, codifica e configurazione dei test. Di conseguenza, il valore misurato è rimasto al di sotto della soglia accettabile. A partire dallo *sprint_e* 12, il gruppo ha esteso la completezza descrittiva integrando commenti nel codice e ampliando il documento di *Specificazione Tecnica*. Questo ha permesso al team di ottimizzare le fasi di testing e manutenzione del codice.

4.15 M.PD.6 – Browser supportati

Tabella dei browser supportati – Ultimo aggiornamento: 2024-09-19

Browser	Supportato
Google Chrome	Sì
Mozilla Firefox	Sì
Safari	Sì
Opera	Sì
Microsoft Edge	Sì
Percentuale di supporto	100%

Tabella 4.2: Tabella dei browser supportati

Nel *Manuale Utente* sono riportate le versioni supportate per ciascun browser.

4.16 M.PD.11 - Code coverage

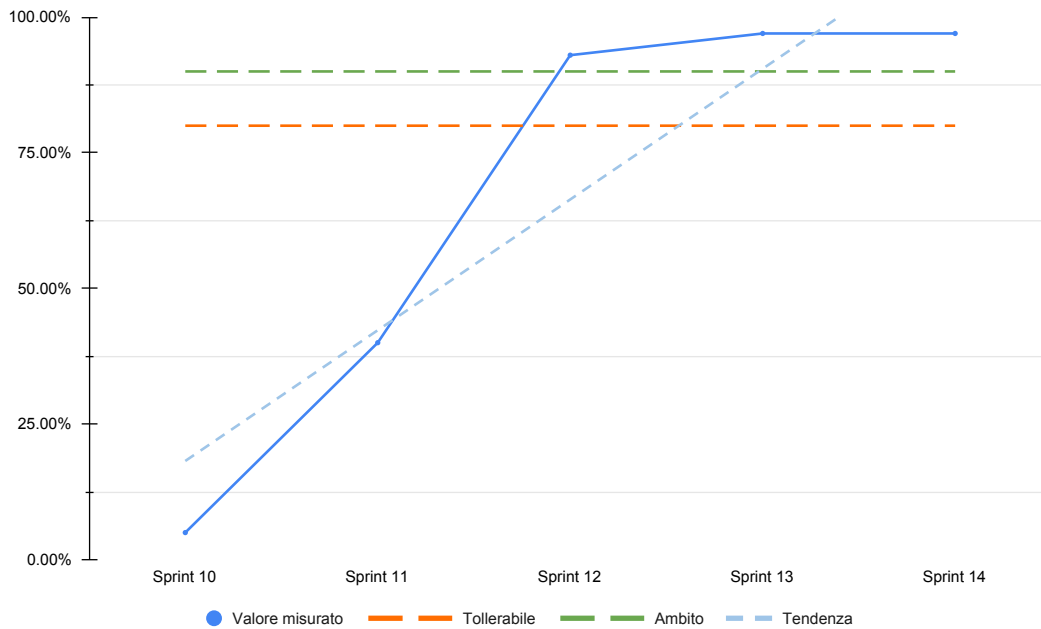


Figura 4.15: M.PD.11 - Code coverage

Nei primi due sprint della PB_e , la maggior parte delle risorse è stata dedicata alla scelta del modello architetturale e alla progettazione del software. Di conseguenza, il team ha dato priorità alla codifica, concentrandosi sulla definizione di una struttura solida per il codice sorgente. Durante la seconda metà dello *sprint*_e 11, il gruppo ha riscontrato difficoltà nella configurazione dell'ambiente di test; questo ha ostacolato l'esecuzione parallela delle attività di codifica e testing. Nonostante ciò, il team ha organizzato incontri per approfondire l'uso degli strumenti di testing, al fine di avviare i test con il minor ritardo possibile. Tale approccio ha permesso al gruppo di implementare un buon numero di test sia per il *front-end*_e che per il *back-end*_e entro la fine dello sprint 11. A partire dal dodicesimo sprint, le attività di codifica e testing sono state svolte in parallelo, raggiungendo una copertura del codice superiore al valore ambito.

4.17 M.PD.12 - Adeguatezza delle funzioni sviluppate

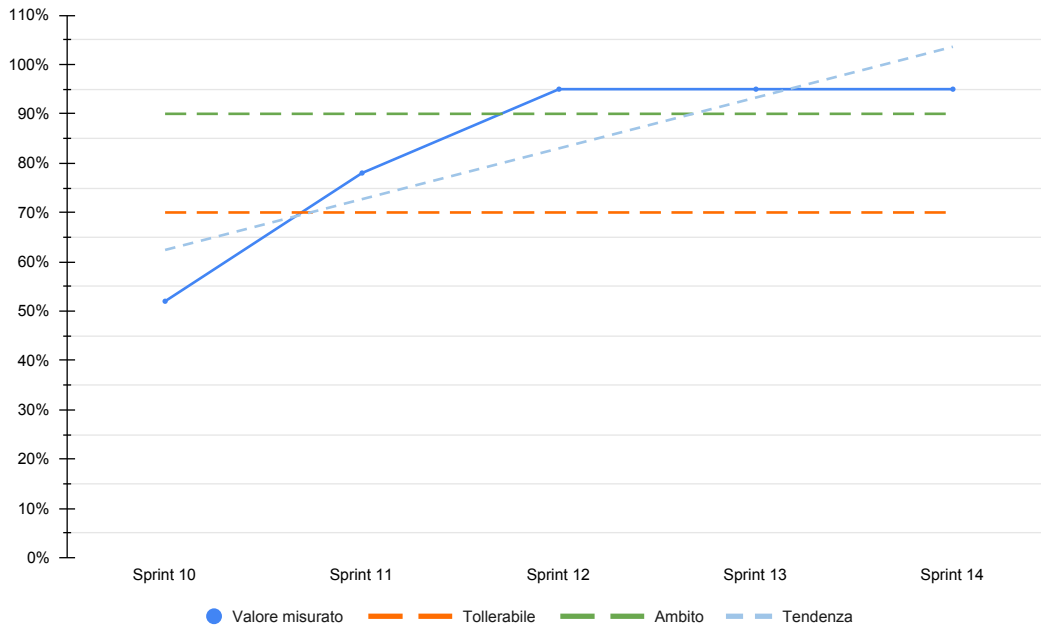


Figura 4.16: M.PD.12 - Adeguatezza delle funzioni sviluppate

L'adeguatezza di una funzione si basa su tre fattori principali: conformità ai requisiti, copertura del codice e superamento dei test. Nel primo sprint della PB_{ϵ} il valore misurato ha superato il 50%, in quanto il codice *front-end* _{ϵ} aveva già raggiunto un livello di adeguatezza tale da soddisfare gli standard previsti. A partire dallo *sprint* _{ϵ} 11, la percentuale è cresciuta progressivamente, fino a superare la soglia desiderata nel dodicesimo sprint. Durante lo sviluppo, il team ha verificato che per ciascuna funzione:

- La copertura dei test fosse superiore al 90%;
- La percentuale di superamento dei test raggiungesse il 100%;
- Il codice fosse conforme ai requisiti e opportunamente commentato.

Grazie a questi controlli, il team ha assicurato che le funzioni sviluppate fossero in linea con le aspettative.

4.18 M.PD.13 – Linee medie di codice per metodo

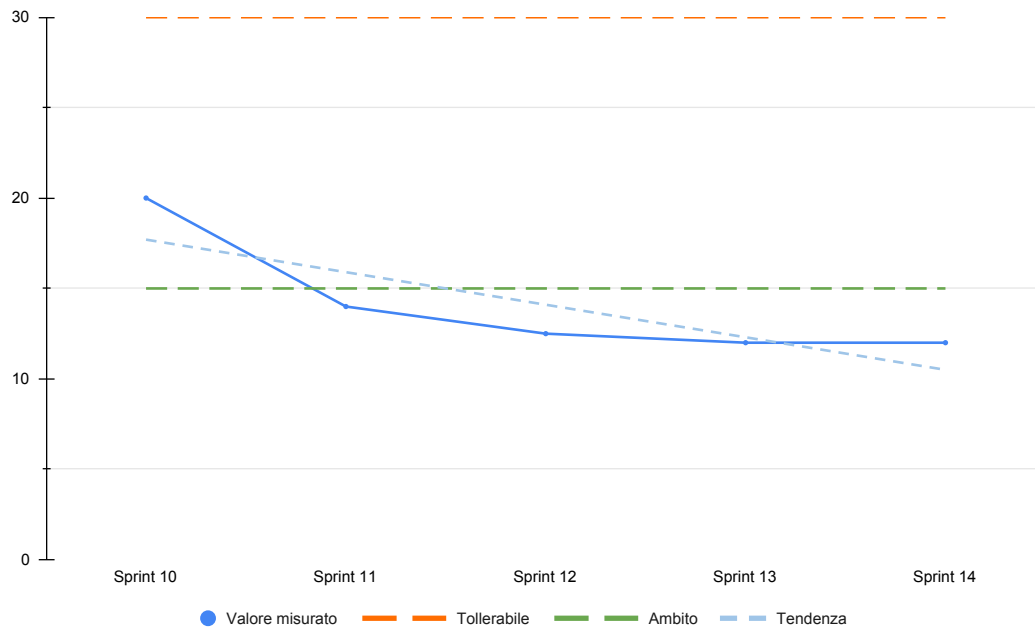


Figura 4.17: M.PD.13 – Linee medie di codice per metodo

Inizialmente, il numero medio di linee di codice per metodo (o funzione) non ha raggiunto la soglia ambita, rimanendo all'interno del range considerato accettabile. Questo risultato è stato influenzato dalla necessità di una fase di studio preliminare per la separazione delle responsabilità dei componenti e per l'implementazione del modello architetturale scelto. Inoltre, il codice del *front-end* era caratterizzato da ridondanze e mancava di una struttura chiara. A partire dall'undicesimo *sprint*, il team ha eseguito un refactoring completo del codice front-end, suddividendo le funzioni più complesse in segmenti più piccoli e gestibili. A seguire, l'implementazione dell'architettura *back-end* ha permesso una chiara separazione delle responsabilità tra i vari metodi. Come mostrato nel grafico, queste modifiche progettuali hanno consentito al team di raggiungere e superare la soglia ambita.

4.19 M.PD.14 - Complessità ciclomatica

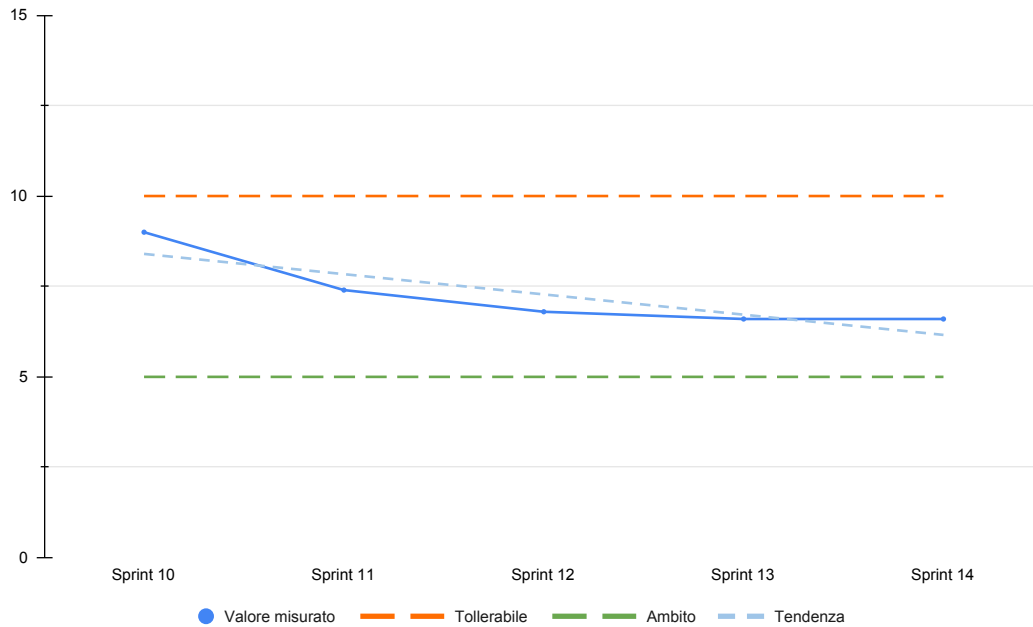


Figura 4.18: M.PD.14 - Complessità ciclomatica

La complessità ciclomatica è rimasta entro i limiti di tollerabilità per tutta la durata della PB_e . Il grafico evidenzia un picco di complessità in corrispondenza dello *sprint*₆, quando la progettazione era ancora in fase preliminare. Con l'avvio del refactoring e l'implementazione del modello architetturale selezionato, il gruppo ha osservato una diminuzione costante della complessità. Questo miglioramento è stato anche conseguenza della rielaborazione delle funzioni più complesse, che sono state suddivise in unità più piccole e con responsabilità limitate. Per quanto riguarda i test, il team è riuscito a mantenere la complessità ciclomatica inferiore a 10, rispettando il range previsto dal *Piano di Qualifica*.

4.20 M.PD.17 – Percentuale di test superati

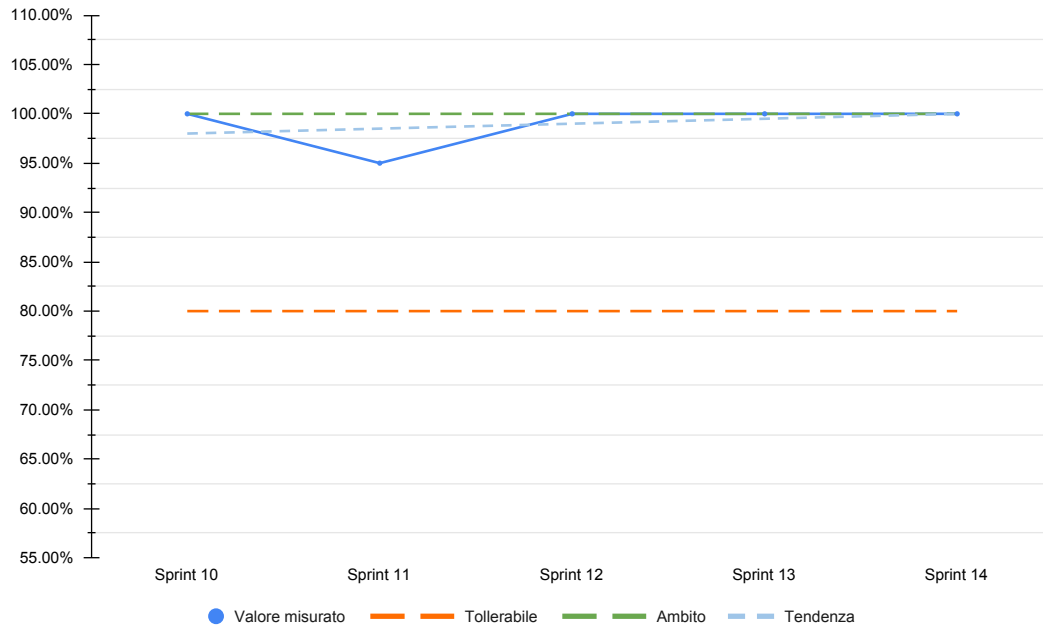


Figura 4.19: M.PD.17 – Percentuale di test superati

A causa dell'inesperienza del team nella scrittura dei test e nell'uso di *mock_e* e *stub_e*, l'undicesimo sprint si è concluso senza il superamento completo dei test. Dal dodicesimo *sprint_e*, le attività di codifica e testing sono state eseguite in parallelo, riducendo significativamente il tasso di errori. Inoltre, il superamento di tutti i test è stato reso una condizione necessaria per chiudere le *pull request_e*. Questo ha garantito che i difetti rilevati venissero risolti prima dell'integrazione nell'ambiente condiviso.

4.21 M.PD.18 – Rimozione dei difetti

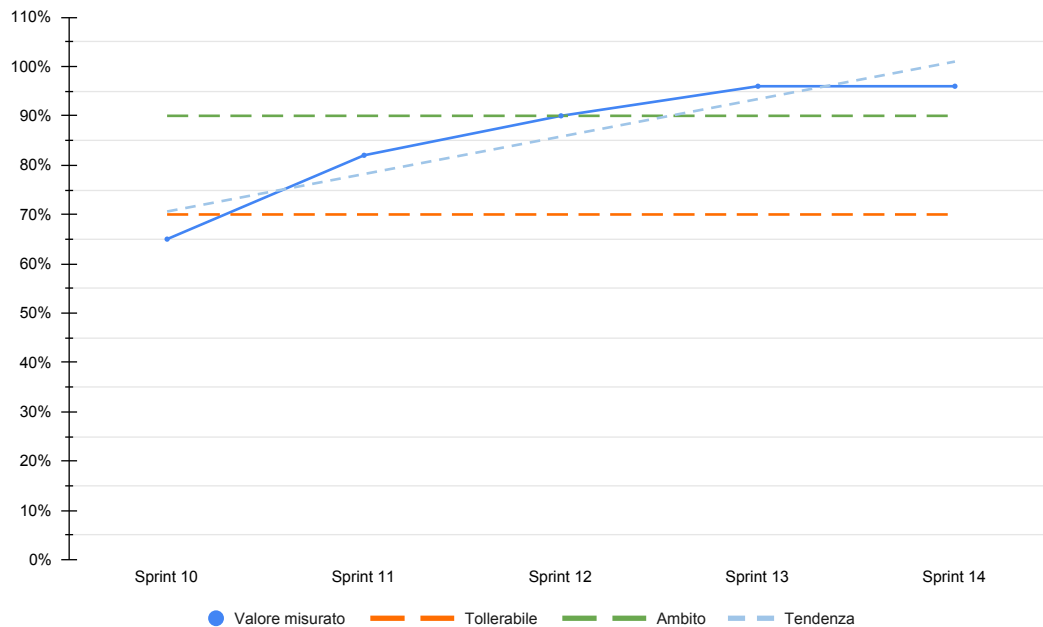


Figura 4.20: M.PD.18 – Rimozione dei difetti

Nel primo sprint della PB_{ϵ} , il team ha configurato SonarCloud, uno strumento di analisi statica del codice finalizzato all'individuazione e alla rimozione dei difetti. A ogni aggiornamento di una $pull\ request_{\epsilon}$, il bot di SonarCloud genera un report che segnala potenziali problemi. L'obiettivo del gruppo era rimuovere tutti i difetti prima della chiusura delle pull request. Tuttavia, alcuni problemi di sicurezza e di affidabilità erano legati a Dockerfile o al template di PrimeVue. Pertanto, il team ha deciso di lasciare in sospeso la risoluzione di tali anomalie, assegnando loro un peso minore nel calcolo della metrica. A partire dallo $sprint_{\epsilon}$ 12, l'implementazione dei test automatici e la configurazione di un workflow su $GitHub_{\epsilon}$ hanno contribuito a ottimizzare il rilevamento dei difetti nel codice. Di conseguenza, la percentuale di rimozione dei difetti è salita progressivamente, superando il valore ambito. La maggior parte delle vulnerabilità rimosse riguardava la duplicazione del codice, che è stata ridotta significativamente. Nello sprint 12, il team ha corretto anche alcuni errori di sicurezza segnalati da SonarCloud.

4.22 M.PD.19 – Tolleranza agli errori

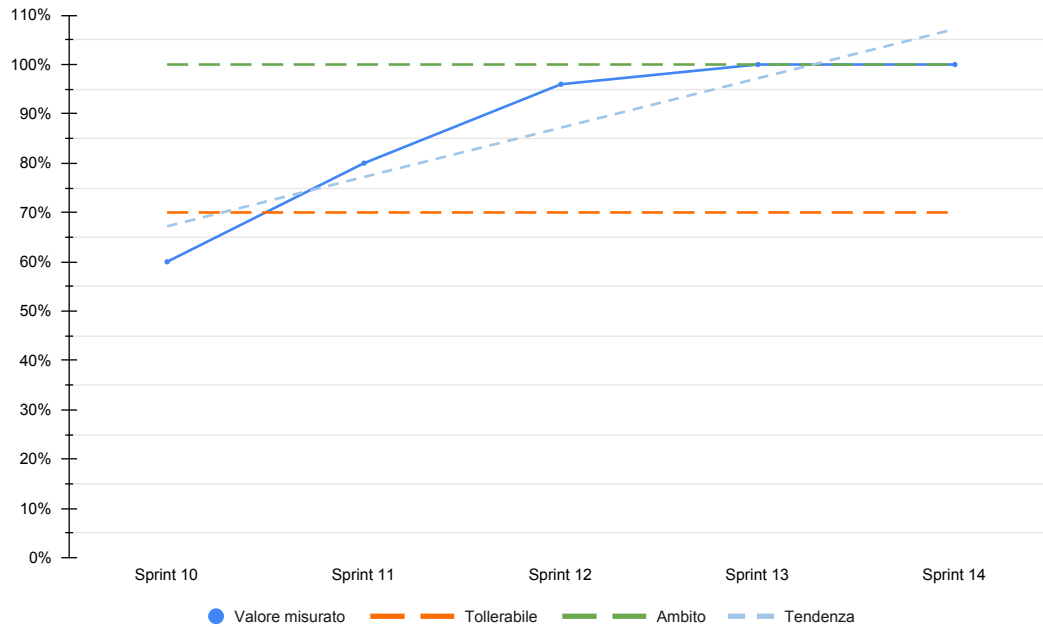


Figura 4.21: M.PD.19 – Tolleranza agli errori

La tolleranza agli errori è stata mantenuta su livelli alti sin dal primo sprint della PB_{ϵ} , grazie all'implementazione del costrutto try-catch per gestire le eccezioni nel codice $front-end_{\epsilon}$. Durante lo $sprint_{\epsilon}$ 11, il team ha introdotto una gestione centralizzata degli errori e rafforzato i controlli sui metodi che eseguono chiamate API_{ϵ} . Questo ha consentito di migliorare la tolleranza, rientrando nel range di tollerabilità previsto. Successivamente, sono state sviluppate le funzionalità mancanti per la gestione degli errori, in particolare quelle relative alle operazioni $CRUD_{\epsilon}$ sui $dizionari\ dati_{\epsilon}$. I test hanno inoltre evidenziato sezioni di codice che necessitavano dell'uso del costrutto try-catch. Dopo aver mitigato gli errori meno evidenti rilevati dai test, il team ha raggiunto una tolleranza del 100%.

4.23 M.PD.20 - Impatto delle modifiche



Figura 4.22: M.PD.20 - Impatto delle modifiche

Il team ha stabilito che ogni modifica, prima di essere integrata nell'ambiente condiviso, deve superare i seguenti controlli:

- Analisi di SonarCloud;
- Test automatici definiti nel workflow;
- Revisione da parte del verificatore.

L'obiettivo del team era garantire che il codice introdotto nel *repository*₆ non compromettesse il funzionamento generale dell'applicazione. Tuttavia, durante l'undicesimo e il dodicesimo *sprint*₆, alcune modifiche hanno avuto un impatto negativo a causa di una copertura del codice insufficiente e della mancanza di test di sistema. Dopo aver raggiunto un livello adeguato di copertura, ogni *pull request*₆ è stata sottoposta a un numero di controlli sufficiente per garantire la solidità delle modifiche associate. In caso di errori durante la fase di verifica, il gruppo ha prontamente rimosso difetti e vulnerabilità.

4.24 Altre metriche del codice

Metriche del codice – Ultimo aggiornamento: 2024-09-19

Metriche del codice	Sprint 10	Sprint 11	Sprint 12	Sprint 13	Sprint 14
M.PD.15 – Duplicazione del codice	3% – 5%	< 3%	< 3%	< 3%	< 3%
M.PD.16 – Indice di manutenibilità	B	A	A	A	A
M.PD.7 – Click Depth	1	1	1	1	1
M.PD.8 – Ampiezza	3	2	2	2	2
M.PD.10 – Tempo di risposta	9	7	7	7	7

Tabella 4.3: Metriche del codice

Durante il primo sprint della PB_{ϵ} , la quantità di codice ripetuto all'interno del progetto è rimasta inferiore al 5%, agevolando il lavoro di refactoring. Con l'implementazione del modello architetturale scelto, il gruppo ha raggiunto l'obiettivo desiderato, portando la duplicazione del codice a meno del 3%. La struttura del progetto e la definizione di molteplici interfacce hanno permesso al team di distribuire le responsabilità, minimizzando le ridondanze. A partire dallo sprint 11, SonarCloud ha assegnato al gruppo una A come indice di manutenibilità. Questo risultato dimostra che le scelte progettuali e architetturali del team sono state conformi agli standard qualitativi previsti.

Per migliorare l'esperienza utente, il team ha mantenuto una click depth pari a 1, rendendo più semplice la navigazione tra le pagine. L'ampiezza del menu di navigazione è stata fissata a 2, con l'obiettivo di ridurre il sovraccarico cognitivo. Questi valori rispettano il modello di una struttura "ampia e poco profonda".

Il tempo medio di risposta oscilla tra 5 e 10 secondi. Questo valore tiene in considerazione il tempo necessario per la creazione degli $indici_{\epsilon}$ e i ritardi dovuti alla *ricerca semantica* _{ϵ} . Il miglioramento tra lo sprint 10 e 11 è attribuibile all'implementazione del modello architetturale e all'introduzione di accorgimenti per migliorare le prestazioni.