

# Piano di qualifica

v0.11



7Last



## Versioni

Ver.	Data	Autore	Verificatore	Descrizione
0.11	2024-05-23	Antonio Benetazzo	Matteo Tiozzo	Ridefinite le metriche di qualità
0.10	2024-05-23	Elena Ferro	Davide Malgarise	Aggiunti Test di Unità e Test di Integrazione
0.9	2024-05-08	Matteo Tiozzo	Leonardo Baldo	Popolati grafici cruscotto e aggiornamento automiglioramento
0.8	2024-05-02	Antonio Benetazzo	Davide Malgarise	Aggiunta sezione automiglioramento
0.7	2024-05-02	Antonio Benetazzo	Davide Malgarise	Cruscotto di valutazione della qualità
0.6	2024-04-30	Antonio Benetazzo	Davide Malgarise	Aggiunti testi introduttivi dei test
0.5	2024-04-22	Antonio Benetazzo	Davide Malgarise	Revisione e piccole correzioni
0.4	2024-04-16	Elena Ferro	Valerio Occhinegro	Riordinamento obbiettivi di qualità
0.3	2024-04-07	Valerio Occhinegro	Matteo Tiozzo	Stesura documento
0.2	2024-03-29	Matteo Tiozzo	Elena Ferro	Modificato tabella versioni
0.1	2024-03-28	Valerio Occhinegro	Matteo Tiozzo	Prima redazione

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Obiettivo del documento . . . . .	5
1.2	Glossario . . . . .	5
1.3	Riferimenti . . . . .	5
1.3.1	Riferimenti normativi . . . . .	5
1.3.2	Riferimenti informativi . . . . .	5
<b>2</b>	<b>Metriche di qualità per obiettivo</b>	<b>7</b>
2.1	Processi di base e/o primari . . . . .	7
2.1.1	Analisi dei requisiti . . . . .	7
2.1.2	Progettazione . . . . .	8
2.1.2.1	Usabilità . . . . .	8
2.1.3	Fornitura . . . . .	9
2.1.4	Sviluppo . . . . .	10
2.1.4.1	Complessità e struttura del codice . . . . .	10
2.1.4.2	Efficienza . . . . .	12
2.2	Processi di supporto . . . . .	12
2.2.1	Documentazione . . . . .	12
2.2.2	Verifica . . . . .	13
2.2.3	Gestione dei rischi . . . . .	13
2.2.4	Gestione della Qualità . . . . .	14
2.3	Processi organizzativi . . . . .	14
2.3.1	Pianificazione . . . . .	14
<b>3</b>	<b>Metodologie di Testing</b>	<b>16</b>
3.1	Test di Unità . . . . .	16
3.2	Test di Integrazione . . . . .	21
3.3	Test di Sistema . . . . .	23
3.4	Test di Accettazione . . . . .	26
<b>4</b>	<b>Cruscotto di valutazione della qualità</b>	<b>29</b>
4.1	Qualità del processo di Analisi dei requisiti . . . . .	29
4.1.1	1M-CRO - Copertura dei requisiti obbligatori . . . . .	29
4.1.2	2M-CRD - Copertura dei requisiti desiderabili . . . . .	30
4.1.3	3M-CROP - Copertura dei requisiti opzionali . . . . .	31

4.2	Qualità del processo di Fornitura . . . . .	32
4.2.1	9M-EV - Earned Value (EV) e 10M-PV - Planned Value (PV) . . . . .	32
4.2.2	11M-AC - Actual Cost (AC) e 14M-ETC - Estimate to Complete (ETC) . . . . .	33
4.2.3	12M-CV - Cost Variance (CV) e 33M-SV - Schedule Variance (SV) . . . . .	34
4.2.4	13M-EAC - Estimated at Completion (EAC) . . . . .	35
4.3	Qualità del processo di Documentazione . . . . .	36
4.3.1	22M-IG - Indice Gulpease . . . . .	36
4.3.2	23M-CO - Correttezza Ortografica . . . . .	37
4.4	Qualità del processo di Verifica . . . . .	38
4.4.1	24M-CC - Code coverage . . . . .	38
4.4.2	25M-BC - Branch coverage . . . . .	39
4.4.3	26M-SC - Statement coverage . . . . .	40
4.4.4	27M-FD - Failure density . . . . .	41
4.4.5	28M-PTCP - Passed Test Cases Percentage . . . . .	42
4.5	Qualità del processo di Gestione dei rischi . . . . .	43
4.5.1	29M-NCR - Non-Calculated Risk . . . . .	43
4.6	Qualità del processo di Gestione della qualità . . . . .	44
4.6.1	30M-QMS - Quality Metrics Satisfied . . . . .	44
4.6.2	31M-TE - Efficienza Temporale . . . . .	45
4.7	Qualità del processo di Pianificazione . . . . .	46
4.7.1	32M-RSI - Requirements stability index (RSI) . . . . .	46
<b>5</b>	<b>Iniziative di automiglioramento per la qualità</b>	<b>47</b>
5.1	Introduzione . . . . .	47
5.2	Problemi rilevati ed iniziative adottate . . . . .	47
5.3	Considerazioni finali . . . . .	48

## Indice delle tabelle

1	Metriche di <i>Analisi dei Requisiti</i> <sub>G</sub> . . . . .	8
2	Metriche di Progettazione - Usabilità . . . . .	9
3	Metriche di Fornitura . . . . .	10
4	Metriche di Sviluppo - Complessità e struttura del codice . . . . .	11
5	Metriche di Sviluppo - Efficienza . . . . .	12
6	Metriche di Documentazione . . . . .	12
7	Metriche di Verifica . . . . .	13

8	Metriche di Gestione dei processi . . . . .	14
9	Metriche di Gestione della Qualità . . . . .	14
10	Metriche di Pianificazione . . . . .	15
11	Test di Unità . . . . .	21
12	Test di Integrazione . . . . .	23
13	Test di Sistema . . . . .	26
14	Test di Accettazione . . . . .	28

## Indice delle immagini

1	Percentuale di copertura dei requisiti obbligatori . . . . .	29
2	Percentuale di copertura dei requisiti desiderabili . . . . .	30
3	Percentuale di copertura dei requisiti opzionali . . . . .	31
4	Proiezione del PV e dell'EV . . . . .	32
5	Proiezione dell'AC e dell'ETC . . . . .	33
6	Andamento percentuale di SV e CV . . . . .	34
7	Proiezione dell'EAC . . . . .	35
8	Andamento indice di Gulpease per ciascun documento . . . . .	36
9	Errori ortografici per ciascun documento . . . . .	37
10	Percentuale di code coverage dei test implementati . . . . .	38
11	Percentuale di branch coverage dei test implementati . . . . .	39
12	Percentuale di statement coverage dei test implementati . . . . .	40
13	Percentuale di failure density . . . . .	41
14	Percentuale di casi di test superati . . . . .	42
15	Rischi non calcolati occorsi durante il progetto . . . . .	43
16	Percentuale di metriche di qualità soddisfatte . . . . .	44
17	Andamento dell'efficienza temporale . . . . .	45
18	Percentuale di stabilità dei requisiti . . . . .	46



# 1 Introduzione

## 1.1 Obiettivo del documento

Il presente documento ha lo scopo di definire le strategie di verifica e validazione utilizzate per assicurare il corretto funzionamento e uno standard di qualità dello strumento sviluppato e delle attività che lo accompagnano. Sarà sottoposto a revisioni continue, così da prevedere situazioni precedentemente non occorse e da seguire l'evoluzione del progetto.

## 1.2 Glossario

Il glossario<sub>G</sub> è uno strumento utilizzato per risolvere eventuali dubbi riguardanti alcuni termini specifici utilizzati nella redazione del documento. Esso conterrà la definizione dei termini evidenziati e sarà consultabile al seguente [link](#). I termini presenti in tale documento saranno evidenziati da una 'G' a pedice.

## 1.3 Riferimenti

### 1.3.1 Riferimenti normativi

- Regolamento del progetto  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>.
- Norme di progetto  
<https://7last.github.io/docs/rtb/documentazione-interna/norme-di-progetto>

### 1.3.2 Riferimenti informativi

- Standard ISO/IEC 25010  
<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- Standard ISO/IEC 12207:1995  
[https://en.wikipedia.org/wiki/ISO/IEC\\_12207](https://en.wikipedia.org/wiki/ISO/IEC_12207)
- Qualità di prodotto  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T7.pdf>
- Qualità di processo  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T8.pdf>



- Verifica e validazione
  - Introduzione  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T9.pdf>
  - Analisi statica  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T10.pdf>
  - Analisi dinamica  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T11.pdf>
- Capitolato d'appalto C6: SyncCity - A smart city monitoring platform  
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C6.pdf>
- Verbali esterni  
<https://7last.github.io/docs/category/verbali-esterni>
- Verbali interni  
<https://7last.github.io/docs/category/verbali-interni>
- Analisi dei requisiti  
<https://7last.github.io/docs/rtb/documentazione-esterna/analisi-dei-requisiti>
- Glossario  
<https://7last.github.io/docs/rtb/documentazione-interna/glossario>



## 2 Metriche di qualità

La qualità di processo è un criterio fondamentale ed è alla base di ogni prodotto che rispecchi lo stato dell'arte. Per raggiungere tale obiettivo è necessario sfruttare delle pratiche rigorose che consentano lo svolgimento di ogni attività in maniera ottimale. Al fine di valutare nel miglior modo possibile la qualità del prodotto e l'efficacia dei processi, sono state definite delle metriche, meglio specificate nel documento *Norme di Progetto<sub>G</sub>*. Il contenuto di questa sezione è necessario per identificare i parametri che le metriche devono rispettare per essere considerate accettabili o ottime. Esse sono state suddivise utilizzando lo **standard ISO/IEC 12207:1995**, il quale separa i processi di ciclo di vita del software, in tre categorie:

- processi di base e/o primari;
- processi di supporto;
- processi organizzativi.

### 2.1 Processi di base e/o primari

#### 2.1.1 Fornitura

Nella fase di fornitura si definiscono le procedure e le risorse (economiche e temporali) necessarie per la consegna del prodotto. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

#### 1M-PV Planned Value

- **Definizione:** il *Planned Value* (o Valore Pianificato) rappresenta il valore del lavoro programmato per essere completato fino a un determinato momento. Si tratta del budget preventivato per lo sprint in corso.
- **Come si calcola:**

$$PV = BAC \times LP$$

dove:

- *BAC*: Budget At Completion;
- *LP*: percentuale di lavoro pianificato.





- **Valore ammissibile:**

$$PV \geq 0$$

- **Valore ottimo:**

$$PV \leq BAC$$

## 2M-EV Earned Value

- **Definizione:** l'*Earned Value* (o Valore Guadagnato) rappresenta il valore del lavoro effettivamente completato fino al periodo in analisi.
- **Come si calcola:**

$$EV = BAC \times LC$$

dove:

- *BAC*: Budget At Completion;
- *LC*: percentuale di lavoro completato.

- **Valore ammissibile:**

$$EV \geq 0$$

- **Valore ottimo:**

$$EV \leq EAC \text{ (Estimated At Completion)}$$

## 3M-AC Actual Cost

- **Definizione:** l'*Actual Cost* (o Costo Effettivo) rappresenta il costo effettivamente sostenuto per completare il lavoro fino al periodo in analisi.
- **Come si calcola:** si ottiene sommando tutti i costi effettivi sostenuti fino a quella data.
- **Valore ammissibile:**

$$AC \geq 0$$

- **Valore ottimo:**

$$AC \leq EAC$$



#### 4M-SV Schedule Variance

- **Definizione:** la *Schedule Variance* (o Variazione di Programma) rappresenta la differenza tra il valore del lavoro effettivamente completato e il valore del lavoro pianificato, calcolata in percentuale.

- **Come si calcola:**

$$SV = \frac{EV - PV}{EV}$$

- **Valore ammissibile:**

$$SV \geq -10\%$$

- **Valore ottimo:**

$$SV \geq 0\%$$

#### 5M-CV Cost Variance

- **Definizione:** la *Cost Variance* (o Variazione dei Costi) rappresenta la differenza tra il valore del lavoro effettivamente completato e il costo effettivamente sostenuto per completarlo, calcolata in percentuale.

- **Come si calcola:**

$$CV = \frac{EV - AC}{EV}$$

- **Valore ammissibile:**

$$CV \geq -10\%$$

- **Valore ottimo:**

$$CV \geq 0\%$$

#### 6M-CPI Cost Performance Index

- **Definizione:** il *Cost Performance Index* rappresenta il rapporto tra il valore del lavoro effettivamente completato e i costi sostenuti per completarlo.

- **Come si calcola:**

$$CPI = \frac{EV}{AC}$$

- **Valore ammissibile:**

$$CPI \geq 0.8$$



- **Valore ottimo:**

$$CPI \geq 1$$

### 7M-SPI Schedule Performance Index

- **Definizione:** lo *Schedule Performance Index* rappresenta l'efficienza con cui il progetto sta rispettando il programma.

- **Come si calcola:**

$$SPI = \frac{EV}{PV}$$

- **Valore ammissibile:**

$$SPI \geq 0.8$$

- **Valore ottimo:**

$$SPI \geq 1$$

### 8M-EAC Estimate At Completion

- **Definizione:** l'*Estimate at Completion* (o Stima al Completamento) rappresenta una previsione aggiornata del costo totale del progetto basata sulle performance attuali, calcolata in base ai costi effettivamente sostenuti e ai costi stimati per completare il lavoro rimanente.

- **Come si calcola:**

$$EAC = AC + ETC$$

- **Valore ammissibile:**

$$EAC \leq BAC + 5\%$$

- **Valore ottimo:**

$$EAC \leq BAC$$

### 9M-ETC Estimate To Complete

- **Definizione:** l'*Estimate to Complete* (o Stima al Completamento) rappresenta una previsione del costo necessario per completare le attività rimanenti del progetto basata sulle performance attuali.



- **Come si calcola:**

$$ETC = EAC - AC$$

- **Valore ammissibile:**

$$ETC \geq 0$$

- **Valore ottimo:**

$$ETC \leq EAC$$

## 10M-OTDR On-Time Delivery Rate

- **Definizione:** l'*On-Time Delivery Rate* (o Tasso di Consegna nei Tempi) rappresenta la percentuale di attività completate entro la data di scadenza.

- **Come si calcola:**

$$OTDR = \frac{AP}{AT}$$

dove:

- *AP*: attività completate entro la data di scadenza;
- *AT*: attività totali.

- **Valore ammissibile:**

$$OTDR \geq 90\%$$

- **Valore ottimo:**

$$OTDR \geq 95\%$$

## 2.1.2 Sviluppo

Nella fase di sviluppo si realizza il prodotto software, seguendo le specifiche definite in fase di progettazione. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

### 2.1.2.1 Analisi dei requisiti

Questa fase consiste nell'esaminare le richieste della proponente<sub>G</sub> e nel definire i requisiti che il prodotto dovrà soddisfare. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.



### 11M-PRO Percentuale Requisiti Obbligatori

- **Definizione:** rappresenta la percentuale di requisiti obbligatori soddisfatti secondo quanto definito nel documento *Analisi dei Requisiti*<sub>G</sub>.

- **Come si calcola:**

$$PRO = \frac{ROS}{ROT}$$

dove:

- *ROS*: Requisiti Obbligatori Soddisfatti;
- *ROT*: Requisiti Obbligatori Totali.

- **Valore ammissibile:**

$$PRO \geq 100\%$$

- **Valore ottimo:**

$$PRO = 100\%$$

### 12M-PRD Percentuale Requisiti Desiderabili

- **Definizione:** rappresenta la percentuale di requisiti desiderabili soddisfatti secondo quanto definito nel documento *Analisi dei Requisiti*<sub>G</sub>.

- **Come si calcola:**

$$PRD = \frac{RDS}{RDT}$$

dove:

- *RDS*: Requisiti Desiderabili Soddisfatti;
- *RDT*: Requisiti Desiderabili Totali.

- **Valore ammissibile:**

$$PRD \geq 35\%$$

- **Valore ottimo:**

$$PRD = 100\%$$



### 13M-PRO Percentuale Requisiti Opzionali

- **Definizione:** rappresenta la percentuale di requisiti opzionali soddisfatti secondo quanto definito nel documento *Analisi dei Requisiti<sub>G</sub>*.

- **Come si calcola:**

$$PRO = \frac{ROS}{ROT}$$

dove:

- *ROS*: Requisiti Opzionali Soddisfatti;
- *ROT*: Requisiti Opzionali Totali.

- **Valore ammissibile:**

$$PRO \geq 0\%$$

- **Valore ottimo:**

$$PRO \geq 100\%$$

### 2.1.2.2 Codifica

Queste metriche aiutano a valutare la qualità del codice, la complessità e la manutenibilità.

### 14M-PPM Parametri Per Metodo

- **Definizione:** numero medio di parametri passati ai metodi. Un numero elevato di parametri può indicare che un metodo è troppo complesso o che potrebbe essere suddiviso in metodi più piccoli.

- **Come si calcola:**

$$PPM = \frac{P}{M}$$

dove:

- *P*: numero totale di parametri;
- *M*: numero totale di metodi.

- **Valore ammissibile:**

$$PPM \leq 7$$

- **Valore ottimo:**

$$PPM \leq 5$$



## 15M-CPC Campi Per Classe

- **Definizione:** numero medio di campi (variabili di istanza) per classe. Un numero elevato di campi dati può indicare che una classe sta facendo troppo e che potrebbe essere suddivisa in classi più piccole.

- **Come si calcola:**

$$CPC = \frac{CD}{CL}$$

dove:

- $CD$ : numero totale di campi dati;
- $CL$ : numero totale di classi.

- **Valore ammissibile:**

$$CPC \leq 8$$

- **Valore ottimo:**

$$CPC \leq 5$$

## 16M-LCPM Linee di Codice Per Metodo

- **Definizione:** numero medio di linee di codice per metodo. Metodi troppo lunghi possono essere difficili da leggere, capire e mantenere.

- **Come si calcola:**

$$LCPM = \frac{LC}{M}$$

dove:

- $LC$ : numero totale di linee di codice;
- $M$ : numero totale di metodi.

- **Valore ammissibile:**

$$LCPM \leq 50$$

- **Valore ottimo:**

$$LCPM \leq 20$$



## 17M-CCM Complessità CicloMatica

- **Definizione:** la *Complessità CicloMatica* rappresenta la complessità di un programma sulla base del numero di percorsi lineari indipendenti attraverso il codice sorgente. Un valore elevato indica un codice più complesso e potenzialmente più difficile da mantenere.

- **Come si calcola:**

$$CCM = E - N + 2P$$

dove:

- $E$ : numero di archi del grafo;
- $N$ : numero di nodi del grafo;
- $P$ : numero di componenti connesse.

- **Valore ammissibile:**

$$CCM \leq 6$$

- **Valore ottimo:**

$$CCM \leq 3$$

## 2.2 Processi di supporto

I processi di supporto si affiancano ai processi primari per garantire il corretto svolgimento delle attività.

### 2.2.1 Documentazione

La documentazione è un aspetto fondamentale per la comprensione del prodotto e per la sua manutenibilità. A livello pratico consiste nella redazione di manuali e documenti tecnici che descrivano il funzionamento del prodotto e le scelte progettuali adottate. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

## 18M-IG Indice Gulpease

- **Definizione:** l'Indice Gulpease è un indice di leggibilità di un testo tarato sulla lingua italiana. Misura la lunghezza delle parole e delle frasi rispetto al numero di lettere.





- **Come si calcola:**

$$IG = 89 + \frac{300 \times F - 10 \times L}{P}$$

dove:

- $F$ : numero totale di frasi nel documento;
- $L$ : numero totale di lettere nel documento;
- $P$ : numero totale di parole nel documento.

- **Valore ammissibile:**

$$IG \geq 60\%$$

- **Valore ottimo:**

$$IG \geq 90\%$$

## 19M-CO Correttezza Ortografica

- **Definizione:** la correttezza ortografica indica la presenza di errori ortografici nei documenti.
- **Come si calcola:** si contano gli errori ortografici presenti nei documenti.
- **Valore ammissibile:**

*0 errori*

- **Valore ottimo:**

*0 errori*

### 2.2.2 Verifica

La verifica è un processo che si occupa di controllare che il prodotto soddisfi i requisiti stabiliti e sia pienamente funzionante. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

## 20M-CC Code Coverage

- **Definizione:** la *Code Coverage* indica quale percentuale del codice sorgente è stata eseguita durante i test. Serve per capire quanto del codice è stato verificato dai test automatizzati.



- **Come si calcola:**

$$CC = \frac{LE}{LT}$$

dove:

- *LE*: linee di codice eseguite;
- *LT*: linee di codice totali.

- **Valore ammissibile:**

$$CC \geq 80\%$$

- **Valore ottimo:**

$$CC = 100\%$$

## 21M-BC Branch Coverage

- **Definizione:** la *Branch Coverage* indica quale percentuale dei rami decisionali (percorsi derivanti da istruzioni condizionali come *if*, *for*, *while*) del codice è stata eseguita durante i test.

- **Come si calcola:**

$$BC = \frac{BE}{BT}$$

dove:

- *BE*: rami eseguiti;
- *BT*: rami totali.

- **Valore ammissibile:**

$$BC \geq 80\%$$

- **Valore ottimo:**

$$BC = 100\%$$

## 22M-SC Statement Coverage

- **Definizione:** la *Statement Coverage* indica quale percentuale di istruzioni del codice è stata eseguita durante i test.



- **Come si calcola:**

$$SC = \frac{IE}{IT}$$

dove:

- *IE*: istruzioni eseguite;
- *IT*: istruzioni totali.

- **Valore ammissibile:**

$$SC \geq 80\%$$

- **Valore ottimo:**

$$SC = 100\%$$

### 23M-FD Failure Density

- **Definizione:** la *Failure Density* indica il numero di difetti trovati in un software o in una parte di esso durante il ciclo di sviluppo rispetto alla dimensione del software stesso.

- **Come si calcola:**

$$FD = \frac{DF}{LT}$$

dove:

- *DF*: difetti trovati;
- *LT*: linee di codice totali.

- **Valore ammissibile:**

$$FD \leq 15\%$$

- **Valore ottimo:**

$$FD = 0\%$$

### 24M-PTCP Passed Test Case Percentage

- **Definizione:** la *Passed Test Case Percentage* indica la percentuale di test che sono stati eseguiti con successo su una base di test.



- **Come si calcola:**

$$PTCP = \frac{TS}{TT}$$

dove:

- *TS*: test superati;
- *TT*: test totali.

- **Valore ammissibile:**

$$PTCP \geq 90\%$$

- **Valore ottimo:**

$$PTCP = 100\%$$

### 2.2.3 Risoluzione dei problemi

La risoluzione dei problemi è un processo che mira a identificare, analizzare e risolvere le varie problematiche che possono emergere durante lo sviluppo. La gestione dei rischi, in particolare, si occupa di identificare, analizzare e gestire i rischi che possono insorgere durante lo svolgimento del progetto. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

#### 25M-RMR Risk Mitigation Rate

- **Definizione:** la *Risk Mitigation Rate* indica la percentuale di rischi identificati che sono stati mitigati con successo.

- **Come si calcola:**

$$RMR = \frac{RM}{RT}$$

dove:

- *RM*: rischi mitigati;
- *RT*: rischi totali identificati.

- **Valore ammissibile:**

$$RMR \geq 80\%$$

- **Valore ottimo:**

$$RMR = 100\%$$



## 26M-NCR Rischi Non Calcolati

- **Definizione:** indica il numero di rischi occorsi che non sono stati preventivati durante l'analisi dei rischi.
- **Come si calcola:** si contano i rischi occorsi e non preventivati.
- **Valore ammissibile:**

$$\leq 3$$

- **Valore ottimo:**

$$= 0$$

## 2.2.4 Gestione della Qualità

La gestione della qualità è un processo che si occupa di definire una metodologia per garantire la qualità del prodotto. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

## 27M-FU Facilità di Utilizzo

- **Definizione:** rappresenta il livello di usabilità del prodotto software mediante il numero di errori riscontrati durante l'utilizzo del prodotto da parte di un utente generico.
- **Come si calcola:** si contano gli errori riscontrati durante l'utilizzo del prodotto da parte di un utente che non ha conoscenze pregresse sul prodotto software.
- **Valore ammissibile:**

$$\leq 3 \text{ errori}$$

- **Valore ottimo:**

$$= 0 \text{ errori}$$

## 28M-TA Tempo di Apprendimento

- **Definizione:** indica il tempo massimo richiesto da parte di un utente generico per apprendere l'utilizzo del prodotto.
- **Come si calcola:** si misura il tempo necessario per apprendere l'utilizzo del prodotto da parte di un utente che non ha conoscenze pregresse sul prodotto software.



- **Valore ammissibile:**

$$\leq 12 \text{ minuti}$$

- **Valore ottimo:**

$$\leq 7 \text{ minuti}$$

### 29M-TR Tempo di Risposta

- **Definizione:** indica il tempo massimo di risposta del sistema sotto carico rilevato.
- **Come si calcola:** si misura il tempo massimo necessario per ottenere una risposta dal sistema.

- **Valore ammissibile:**

$$\leq 8 \text{ s}$$

- **Valore ottimo:**

$$\leq 4 \text{ s}$$

### 30M-TE Tempo di Elaborazione

- **Definizione:** indica il tempo massimo di elaborazione di un dato grezzo fino alla sua presentazione rilevato.
- **Come si calcola:** si misura il tempo massimo di elaborazione di un dato grezzo dal momento della sua comparsa nel sistema fino alla sua presentazione all'utente.

- **Valore ammissibile:**

$$\leq 10 \text{ s}$$

- **Valore ottimo:**

$$\leq 5 \text{ s}$$

### 31M-QMS Metriche di Qualità Soddisfatte

- **Definizione:** indica il numero di metriche implementate e soddisfatte, tra quelle definite.



- **Come si calcola:**

$$QMS = \frac{MS}{MT}$$

dove:

- *MS*: metriche soddisfatte;
- *MT*: metriche totali.

- **Valore ammissibile:**

$$QMS \geq 90\%$$

- **Valore ottimo:**

$$QMS = 100\%$$

## 2.3 Processi organizzativi

I processi organizzativi sono processi che si occupano di definire le linee guida e le procedure da seguire per garantire un'efficace gestione e coordinazione del progetto.

### 2.3.1 Pianificazione

La pianificazione è un processo che si occupa di definire le attività da svolgere e le risorse temporali e umane necessarie per il loro svolgimento. Per valutare la qualità di tale processo, sono state definite le seguenti metriche.

#### 32M-RSI Requirements Stability Index

- **Definizione:** il *Requirements Stability Index* (RSI) indica la percentuale di requisiti che sono rimasti invariati rispetto al totale dei requisiti inizialmente definiti. Si tratta di una metrica utilizzata per misurare quanto i requisiti di un progetto rimangono stabili durante il ciclo di vita del progetto stesso, è particolarmente utile per comprendere l'impatto delle modifiche ai requisiti sul progetto.

- **Come si calcola:**

$$RSI = \frac{RI - (RA + RR + RC)}{RI}$$

dove:

- *RI*: requisiti iniziali;
- *RA*: requisiti aggiunti;



- *RR*: requisiti rimossi;
- *RC*: requisiti cambiati.

- **Valore ammissibile:**

$$RSI \geq 75\%$$

- **Valore ottimo:**

$$RSI = 100\%$$





### 3 Metodologie di Testing

In questa sezione verranno illustrate le metodologie di *testing* adottate per garantire il rispetto dei vincoli individuati nella sezione *Requisiti* del documento *Analisi dei Requisiti*<sub>G</sub>. I test verranno suddivisi in cinque categorie:

- test di Unità;
- test di Integrazione;
- test di Sistema;
- test di Regressione;
- test di Accettazione.

Verranno elencate le varie tipologie di test eseguite, indicando il codice del test, una breve descrizione di ciò che viene verificato e lo stato di avanzamento del test, espresso come segue:

- **S**: test superato;
- **NS**: test non superato;
- **NI**: test non implementato.

#### 3.1 Test di Unità

I test di unità verificano il corretto funzionamento delle singole unità di codice, ovvero le più piccole parti di un programma, per assicurarsi che ognuna funzioni correttamente e che sia in grado di eseguire le operazioni richieste.

Codice	Descrizione	Stato
1T-U	Verificare che la classe <code>TemperatureRawData</code> venga creata correttamente.	NI
2T-U	Verificare che il metodo <code>topic()</code> di <code>TemperatureRawData</code> restituisca "temperature".	NI
3T-U	Verificare che il metodo <code>subject()</code> di <code>TemperatureRawData</code> restituisca "temperature-value".	NI



Codice	Descrizione	Stato
4T-U	Verificare che la classe TrafficRawData venga creata correttamente.	NI
5T-U	Verificare che il metodo topic() di TrafficRawData restituisca "traffic".	NI
6T-U	Verificare che il metodo subject() di TrafficRawData restituisca "traffic-value".	NI
7T-U	Verificare che la classe RecyclingPointRawData venga creata correttamente.	NI
8T-U	Verificare che il metodo topic() di RecyclingPointRawData restituisca "recycling_point".	NI
9T-U	Verificare che il metodo subject() di RecyclingPointRawData restituisca "recycling_point-value".	NI
10T-U	Verificare che la classe HumidityRawData venga creata correttamente.	NI
11T-U	Verificare che il metodo topic() di HumidityRawData restituisca "humidity".	NI
12T-U	Verificare che il metodo subject() di HumidityRawData restituisca "humidity-value".	NI
13T-U	Verificare che la classe AirQualityRawData venga creata correttamente.	NI
14T-U	Verificare che il metodo topic() di AirQualityRawData restituisca "air_quality".	NI
15T-U	Verificare che il metodo subject() di AirQualityRawData restituisca "air_quality-value".	NI
16T-U	Verificare che la classe RainRawData venga creata correttamente.	NI
17T-U	Verificare che il metodo topic() di RainRawData restituisca "rain".	NI
18T-U	Verificare che il metodo subject() di RainRawData restituisca "rain-value".	NI
19T-U	Verificare che la classe ChargingStationRawData venga creata correttamente.	NI



Codice	Descrizione	Stato
20T-U	Verificare che il metodo topic() di ChargingStationRawData restituisca "charging_station".	NI
21T-U	Verificare che il metodo subject() di ChargingStationRawData restituisca "charging_station-value".	NI
22T-U	Verificare che la classe ParkingLotRawData venga creata correttamente.	NI
23T-U	Verificare che il metodo topic() di ParkingLotRawData restituisca "parking_lot".	NI
24T-U	Verificare che il metodo subject() di ParkingLotRawData restituisca "parking_lot-value".	NI
25T-U	Verificare che la classe WaterLevelRawData venga creata correttamente.	NI
26T-U	Verificare che il metodo topic() di WaterLevelRawData restituisca "water_level".	NI
27T-U	Verificare che il metodo subject() di WaterLevelRawData restituisca "water_level-value".	NI
28-U	Verificare che il metodo from_str() di SensorType effettui il parsing correttamente.	NI
29-U	Verificare che la classe EnvConfig venga creata correttamente se tutte le variabili d'ambiente sono impostate.	NI
30-U	Verificare che la classe EnvConfig venga creata correttamente se la variabile d'ambiente MAX_BLOCK_MS non è impostata.	NI
31-U	Verificare che la creazione della classe EnvConfig fallisca con un'eccezione se le variabili d'ambiente non sono impostate.	NI
32-U	Verificare che il metodo bootstrap_server della classe EnvConfig ritorni correttamente il valore dell'host concatenato alla porta con ':'..	NI
33-U	Verificare che la classe SensorConfig sia creata correttamente.	NI



Codice	Descrizione	Stato
34-U	Verificare che la creazione della classe SensorConfig fallisca con un'eccezione se il tipo di sensore fornito non esiste.	NI
35-U	Verificare che la creazione della classe SensorConfig fallisca con un'eccezione se il tipo di sensore non è fornito.	NI
36-U	Verificare che la creazione della classe SensorConfig fallisca con un'eccezione se il campo generation_delay non rispetta lo standard ISO8601.	NI
37-U	Verificare che la creazione della classe SensorConfig fallisca con un'eccezione se il campo points_spacing non rispetta lo standard ISO8601.	NI
38-U	Verificare che la funzione simulator_generator() crei correttamente i Simulator a partire da una lista di SensorConfig.	NI
39-U	Verificare che il metodo serialize_temperature_raw_data() serializzi correttamente TemperatureRawData.	NI
40-U	Verificare che il metodo serialize_traffic_raw_data() serializzi correttamente TrafficRawData.	NI
41-U	Verificare che il metodo serialize_recycling_point_raw_data() serializzi correttamente RecyclingPointRawData.	NI
42-U	Verificare che il metodo serialize_humidity_raw_data() serializzi correttamente HumidityRawData.	NI
43-U	Verificare che il metodo serialize_air_quality_raw_data() serializzi correttamente AirQualityRawData.	NI
44-U	Verificare che il metodo serialize_humidity_raw_data() serializzi correttamente HumidityRawData.	NI



Codice	Descrizione	Stato
46-U	Verificare che il metodo <code>serialize_rain_raw_data()</code> serializzi correttamente <code>RainRawData</code> .	NI
47-U	Verificare che il metodo <code>serialize_charging_station_raw_data()</code> serializzi correttamente <code>ChargingStationRawData</code> .	NI
48-U	Verificare che il metodo <code>serialize_parking_lot_raw_data()</code> serializzi correttamente <code>ParkingLotRawData</code> .	NI
49-U	Verificare che il metodo <code>serialize_water_level_raw_data()</code> serializzi correttamente <code>WaterLevelRawData</code> .	NI
50-U	Verificare che il metodo <code>run()</code> della classe <code>Runner</code> esegua correttamente i simulatori.	NI
51-U	Verificare che il metodo <code>stream()</code> della classe <code>TemperatureSimulator</code> generi correttamente i dati casuali.	NI
52-U	Verificare che il metodo <code>stream()</code> della classe <code>TrafficSimulator</code> generi correttamente i dati casuali.	NI
53-U	Verificare che il metodo <code>stream()</code> della classe <code>RecyclingPointSimulator</code> generi correttamente i dati casuali.	NI
54-U	Verificare che il metodo <code>stream()</code> della classe <code>HumiditySimulator</code> generi correttamente i dati casuali.	NI
55-U	Verificare che il metodo <code>stream()</code> della classe <code>AirQualitySimulator</code> generi correttamente i dati casuali.	NI
56-U	Verificare che il metodo <code>stream()</code> della classe <code>RainSimulator</code> generi correttamente i dati casuali.	NI
57-U	Verificare che il metodo <code>stream()</code> della classe <code>ChargingStationSimulator</code> generi correttamente i dati casuali.	NI



Codice	Descrizione	Stato
58-U	Verificare che il metodo <code>stream()</code> della classe <code>ParkingLotSimulator</code> generi correttamente i dati casuali.	NI
59-U	Verificare che il metodo <code>stream()</code> della classe <code>WaterLevelSimulator</code> generi correttamente i dati casuali.	NI

Tabella 1: Test di Unità

### 3.2 Test di Integrazione

I test di integrazione verificano il corretto funzionamento delle interfacce tra le varie unità di codice, assicurandosi che esse interagiscano correttamente tra di loro e che siano in grado di comunicare e scambiarsi i dati necessari.

Codice	Descrizione	Stato
1T-I	Verificare che i dati generati dal sensore di temperatura siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
2T-I	Verificare che i dati generati dal sensore di traffico siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
3T-I	Verificare che i dati generati dal sensore di isola ecologica siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
4T-I	Verificare che i dati generati dal sensore di umidità siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
5T-I	Verificare che i dati generati dal sensore di qualità dell'aria siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
6T-I	Verificare che i dati generati dal sensore di precipitazioni siano pubblicati correttamente nel rispettivo topic Redpanda.	NI



Codice	Descrizione	Stato
7T-I	Verificare che i dati generati dalle colonnine di ricarica siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
8T-I	Verificare che i dati generati dai sensori di occupazione di parcheggi siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
9T-I	Verificare che i dati generati dai sensori di livello dell'acqua siano pubblicati correttamente nel rispettivo topic Redpanda.	NI
10T-I	Verificare che i dati generati dal sensore di temperatura siano memorizzati correttamente nel database.	NI
11T-I	Verificare che i dati generati dal sensore di temperatura aggregati per 5 minuti siano memorizzati correttamente nel database.	NI
12T-I	Verificare che i dati generati dal sensore di temperatura aggregati per settimana siano memorizzati correttamente nel database.	NI
13T-I	Verificare che i dati generati dal sensore di temperatura aggregati per giorno siano memorizzati correttamente nel database.	NI
14T-I	Verificare che i dati generati dal sensore di traffico siano memorizzati correttamente nel database.	NI
15T-I	Verificare che i dati generati dal sensore di traffico aggregati per 5 minuti siano memorizzati correttamente nel database.	NI
16T-I	Verificare che i dati generati dal sensore di traffico aggregati per ora siano memorizzati correttamente nel database.	NI
17T-I	Verificare che i dati generati dal sensore di isola ecologica siano memorizzati correttamente nel database.	NI



Codice	Descrizione	Stato
18T-I	Verificare che i dati generati dal sensore di isola ecologica aggregati per 5 minuti siano memorizzati correttamente nel database.	NI
19T-I	Verificare che i dati generati dal sensore di umidità siano memorizzati correttamente nel database.	NI
20T-I	Verificare che i dati generati dal sensore di qualità dell'aria siano memorizzati correttamente nel database.	NI
21T-I	Verificare che i dati generati dal sensore di precipitazioni siano memorizzati correttamente nel database.	NI
22T-I	Verificare che i dati generati dalle colonnine di ricarica siano memorizzati correttamente nel database.	NI
23T-I	Verificare che i dati generati dai sensori di occupazione di parcheggi siano memorizzati correttamente nel database.	NI
24T-I	Verificare che i dati generati dai sensori di livello dell'acqua siano memorizzati correttamente nel database.	NI
25T-I	Verificare che i dati salvati su Clickhouse siano correttamente accessibili da Grafana.	NI

Tabella 2: Test di Integrazione

### 3.3 Test di Sistema

I test di sistema sono finalizzati alla verifica del soddisfacimento dei requisiti richiesti ed evidenziati nel documento *Analisi dei Requisiti*<sub>G</sub>. Questi test vengono effettuati sul sistema nel suo complesso, per verificare che il software funzioni correttamente e che sia in grado di eseguire le operazioni richieste.





<b>Codice</b>	<b>Descrizione</b>	<b>Stato</b>
<b>1T-S</b>	Verificare che l'accesso al sistema non richieda alcuna procedura di login e che sia direttamente accessibile dall'utente.	NI
<b>2T-S</b>	Verificare che il prodotto non abbia alcuna sezione o funzionalità di amministrazione o gestione riservata.	NI
<b>3T-S</b>	Verificare che i sensori integrati producano una misurazione coerente con il tipo di sensore simulato.	NI
<b>4T-S</b>	Verificare che ogni misurazione inviata dal simulatore contenga l'identificativo del sensore, le misurazioni d'interesse e il timestamp.	NI
<b>5T-S</b>	Verificare che il sistema sia in grado di ricevere e memorizzare correttamente le misurazioni inviate dai sensori.	NI
<b>6T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare la temperatura.	NI
<b>7T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare il traffico.	NI
<b>8T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare il riempimento delle isole ecologiche.	NI
<b>9T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare l'umidità.	NI
<b>10T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare la qualità dell'aria.	NI
<b>11T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare le precipitazioni.	NI
<b>12T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare le colonnine di ricarica.	NI
<b>13T-S</b>	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare l'occupazione dei parcheggi.	NI



Codice	Descrizione	Stato
14T-S	Verificare che il sistema sia in grado di simulare almeno un sensore per rilevare il livello dell'acqua.	NI
14T-S	Verificare che ogni dato generato dai simulatori dei sensori sia strettamente correlato al dato successivo, garantendo una transizione realistica tra le misurazioni.	NI
15T-S	Verificare la facilità di comprensione e l'intuitività dell'interfaccia grafica, garantendo un'esperienza utente piacevole e soddisfacente.	NI
16T-S	Verificare che le dashboard si aggiornino quasi istantaneamente per riflettere i dati provenienti dai sensori entro un massimo di 15 secondi.	NI
17T-S	Verificare che la dashboard del traffico contenga almeno un <i>panel</i> con un grafico time-series.	NI
18T-S	Verificare che la dashboard della temperatura contenga almeno un <i>panel</i> con un grafico time-series.	NI
19T-S	Verificare che la dashboard delle isole ecologiche contenga almeno un <i>panel</i> con un grafico time-series.	NI
20T-S	Verificare che la dashboard dell'umidità contenga almeno un <i>panel</i> con un grafico time-series.	NI
21T-S	Verificare che la dashboard della qualità dell'aria contenga almeno un <i>panel</i> con un grafico time-series.	NI
22T-S	Verificare che la dashboard delle precipitazioni contenga almeno un <i>panel</i> con un grafico time-series.	NI
23T-S	Verificare che la dashboard dei parcheggi contenga almeno un <i>panel</i> con un grafico time-series.	NI
24T-S	Verificare che la dashboard delle colonnine di ricarica contenga almeno un <i>panel</i> con un grafico time-series.	NI



Codice	Descrizione	Stato
25T-S	Verificare che la dashboard del livello di acqua contenga almeno un <i>panel</i> con un grafico time-series.	NI
26T-S	Verificare che la dashboard delle isole ecologiche contenga almeno un <i>panel</i> con un grafico time-series.	NI
27T-S	Verificare che i sensori presenti sulla mappa siano distinguibili in modo chiaro ed inequivocabile, permettendo il riconoscimento della loro tipologia.	NI
28T-S	Verificare che in ciascuna dashboard l'utente possa filtrare la visualizzazione delle misurazioni di uno specifico sensore.	NI
29T-S	Verificare che nella dashboard dei dati grezzi l'utente possa visualizzare la lista delle misurazioni in un formato tabellare, divise per tipo di sensore.	NI
30T-S	Verificare che l'utente riceva notifiche quando i sensori superano pre-determinate soglie di sicurezza.	NI
31T-S	Verificare che l'utente possa visualizzare correttamente le coordinate dei sensori, con un numero congruo di cifre decimali.	NI
32T-S	Verificare che l'utente possa visualizzare correttamente l'unità di misura associata a ciascuna misurazione.	NI
33T-S	Verificare che nella dashboard dei dati grezzi l'utente possa visualizzare una tabella contenente l'identificativo del sensore, la sua tipologia e la data dell'ultimo messaggio da esso inviato.	NI

Tabella 3: Test di Sistema

### 3.4 Test di Accettazione

I test di accettazione vengono effettuati per verificare che il software soddisfi i requisiti richiesti e consentono di ultimare il processo di validazione del prodotto finale. Essi



verranno eseguiti sia dal gruppo di sviluppo *7Last* che dall'azienda proponente *SyncLab S.r.l.*.

Codice	Descrizione	Stato
1T-A	Verificare che tutti i widget relativi alle diverse tipologie di sensori siano visibili sulla dashboard.	NI
2T-A	Verificare che la mappa dei sensori si carichi correttamente e permetta interazioni fluide.	NI
3T-A	Verifica della gestione corretta degli errori nel caso in cui i dati dei sensori non siano disponibili.	NI
4T-A	Verifica della corretta visualizzazione delle misurazioni effettuate nel tempo dai sensori.	NI
6T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di temperatura.	NI
7T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di traffico.	NI
8T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di isola ecologica.	NI
9T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di umidità.	NI
10T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di qualità dell'aria.	NI
11T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di precipitazioni.	NI
12T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di colonnine di ricarica.	NI
13T-A	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di occupazione di parcheggi.	NI



<b>Codice</b>	<b>Descrizione</b>	<b>Stato</b>
<b>14T-A</b>	Verificare che sia possibile visualizzare correttamente la dashboard dei sensori di livello dell'acqua.	NI
<b>15T-A</b>	Verificare che sia possibile visualizzare correttamente la dashboard dei dati grezzi	NI
<b>16T-A</b>	Verificare si possa filtrare correttamente la visualizzazione delle misurazioni in base al sensore che le ha prodotte.	NI
<b>17T-A</b>	Verificare che si possa rimuovere correttamente i filtri attivi per visualizzazione delle misurazioni dei sensori.	NI
<b>18T-A</b>	Verificare che si riceva correttamente una notifica in caso di superamento delle soglie impostate per le misurazioni.	NI

Tabella 4: Test di Accettazione



## 4 Cruscotto di valutazione della qualità

### 4.1 Qualità del processo di Analisi dei requisiti

#### 4.1.1 1M-CRO - Copertura dei requisiti obbligatori

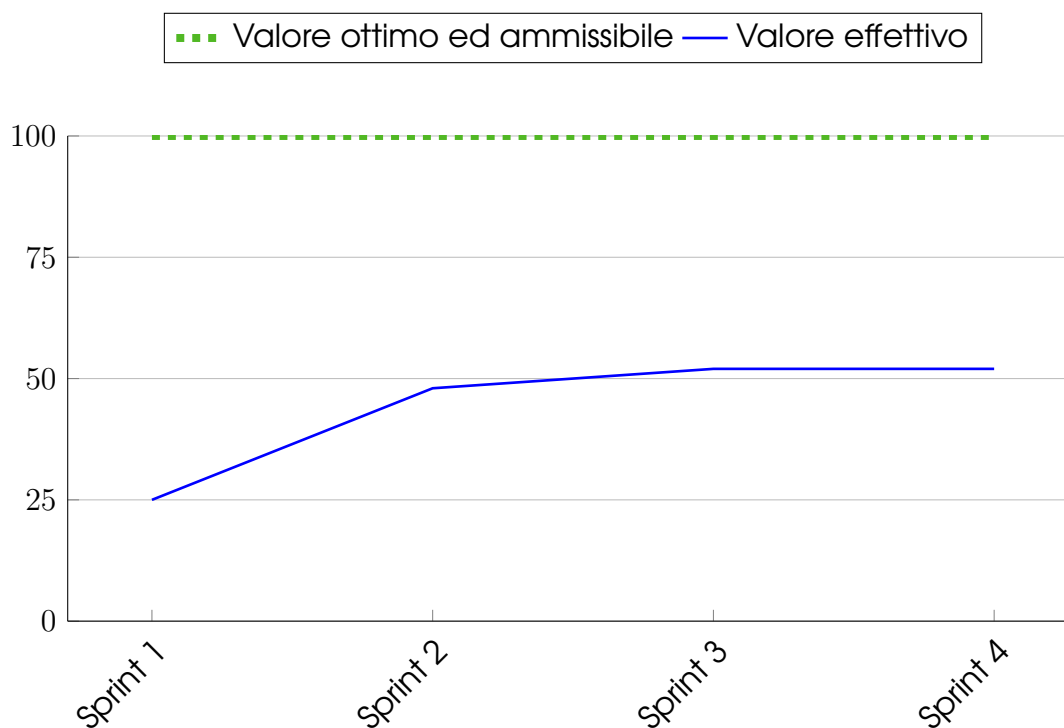


Figure 1: Percentuale di copertura dei requisiti obbligatori



#### 4.1.2 2M-CRD - Copertura dei requisiti desiderabili

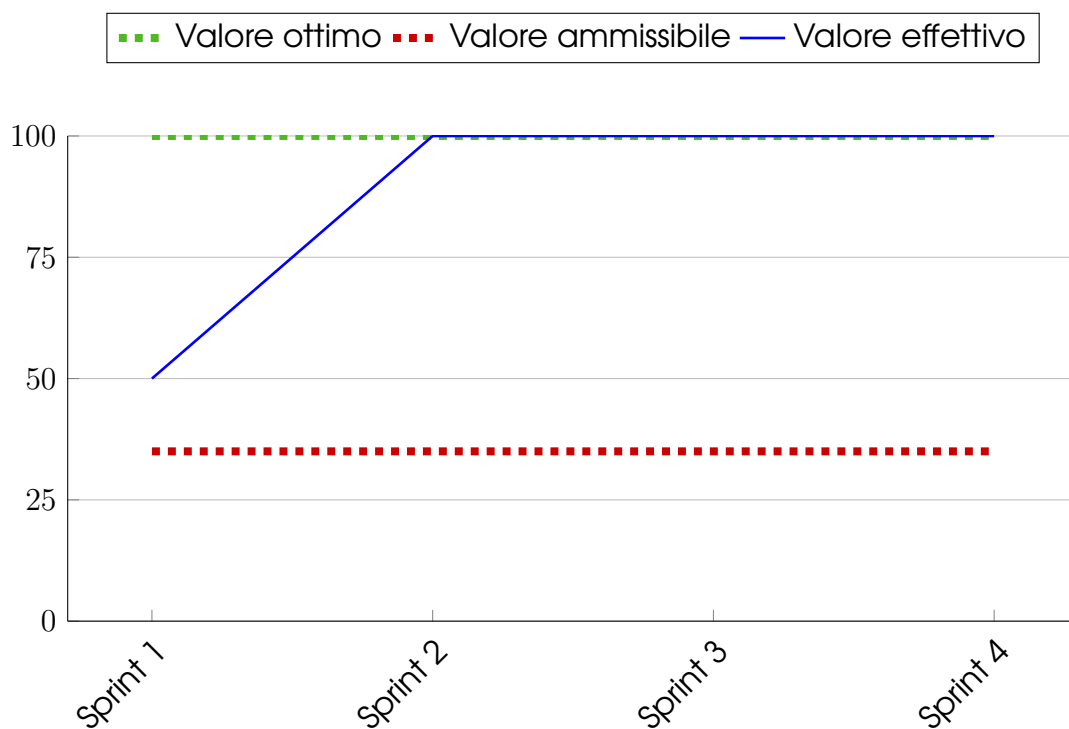


Figure 2: Percentuale di copertura dei requisiti desiderabili



#### 4.1.3 3M-CROP - Copertura dei requisiti opzionali

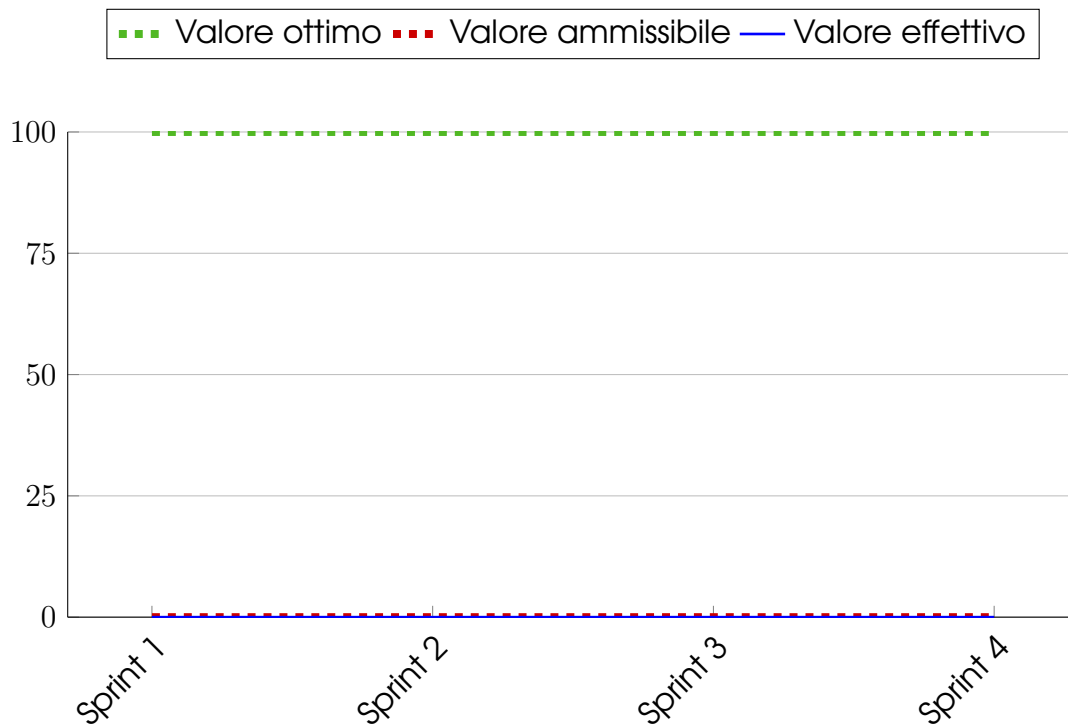


Figure 3: Percentuale di copertura dei requisiti opzionali





## 4.2 Qualità del processo di Fornitura

### 4.2.1 9M-EV - Earned Value (EV) e 10M-PV - Planned Value (PV)

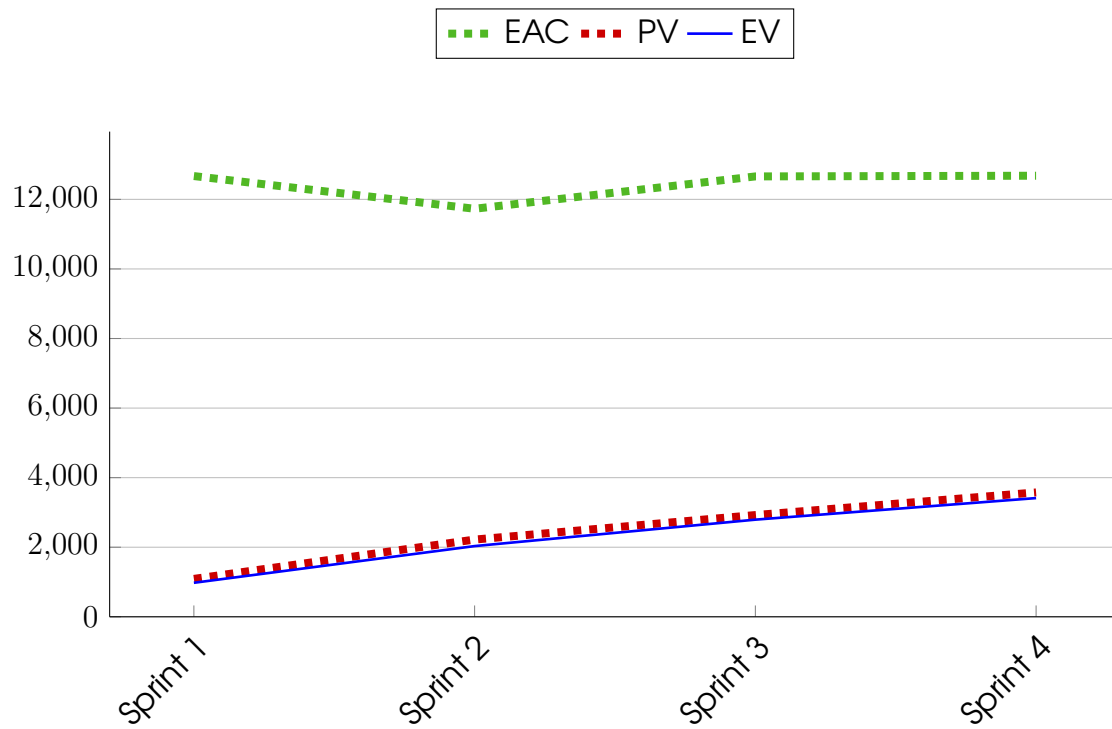


Figure 4: Proiezione del PV e dell'EV



#### 4.2.2 11M-AC - Actual Cost (AC) e 14M-ETC - Estimate to Complete (ETC)

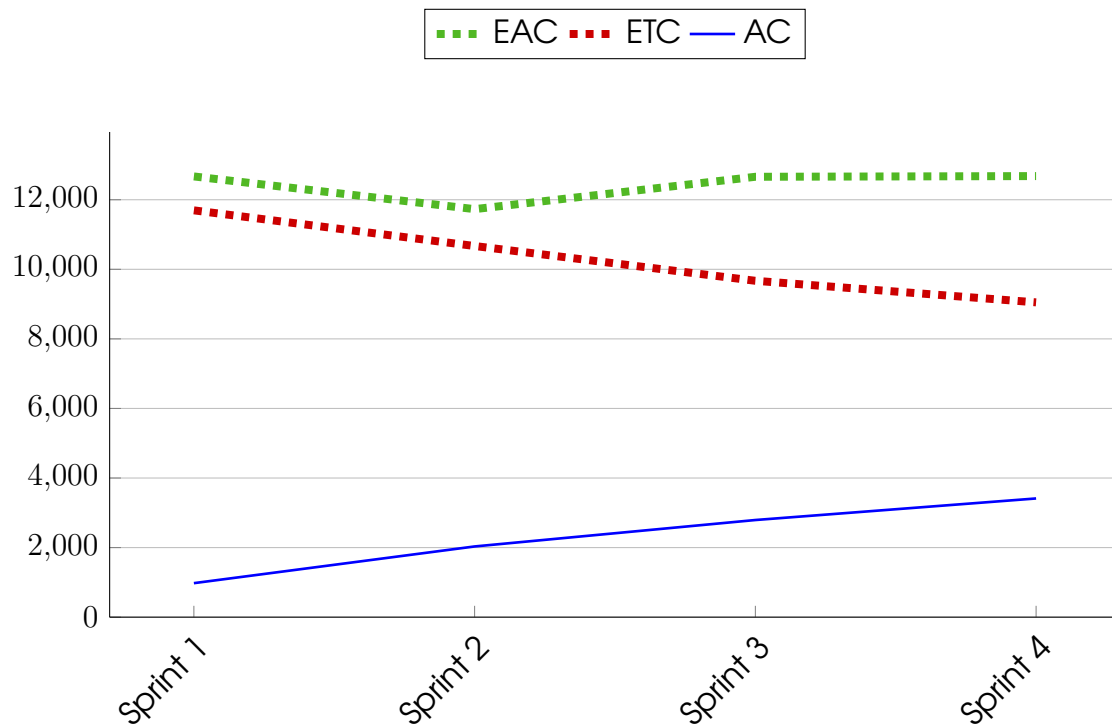


Figure 5: Proiezione dell'AC e dell'ETC



#### 4.2.3 12M-CV - Cost Variance (CV) e 33M-SV - Schedule Variance (SV)

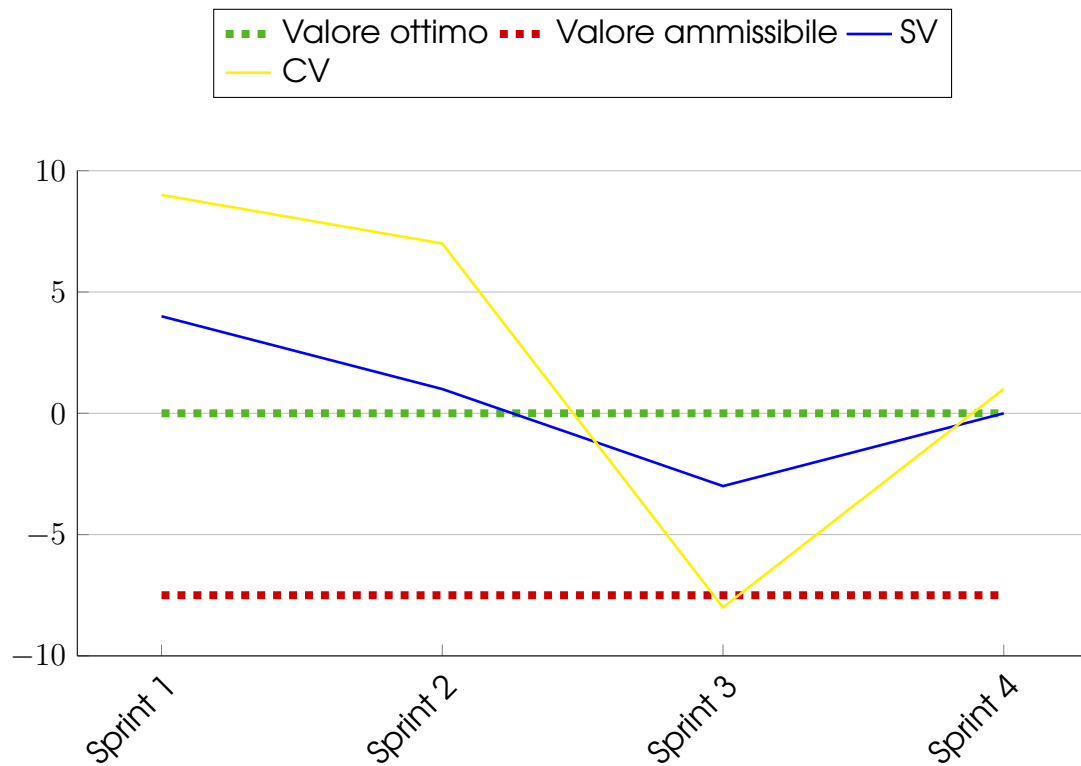


Figure 6: Andamento percentuale di SV e CV



#### 4.2.4 13M-EAC - Estimated at Completion (EAC)

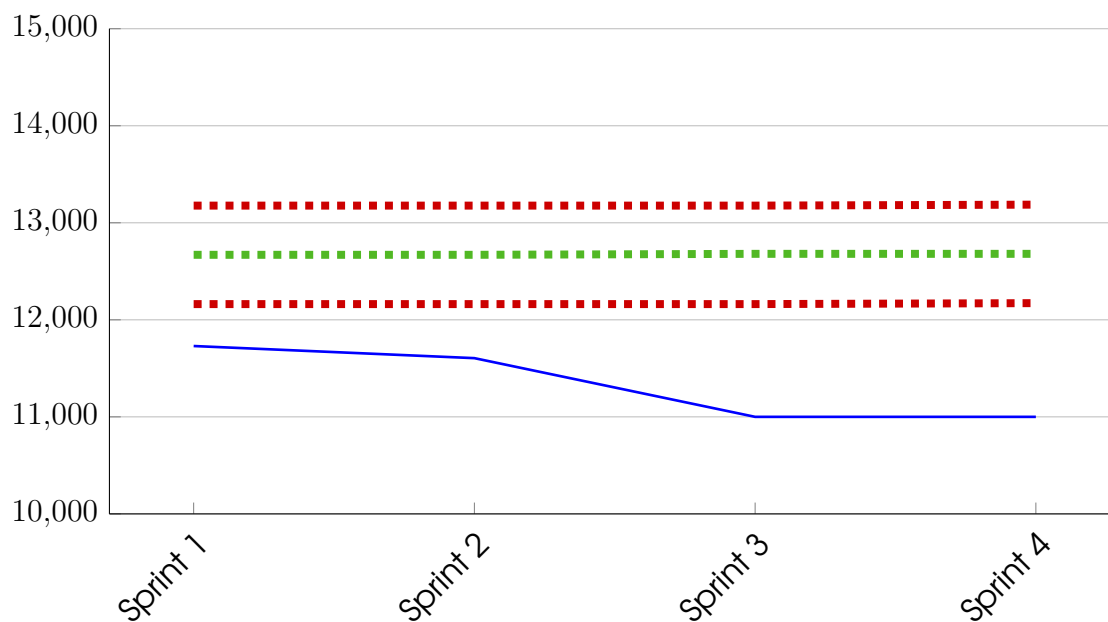
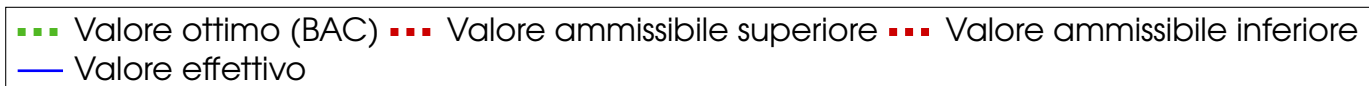


Figure 7: Proiezione dell'EAC



## 4.3 Qualità del processo di Documentazione

### 4.3.1 22M-IG - Indice Gulpease

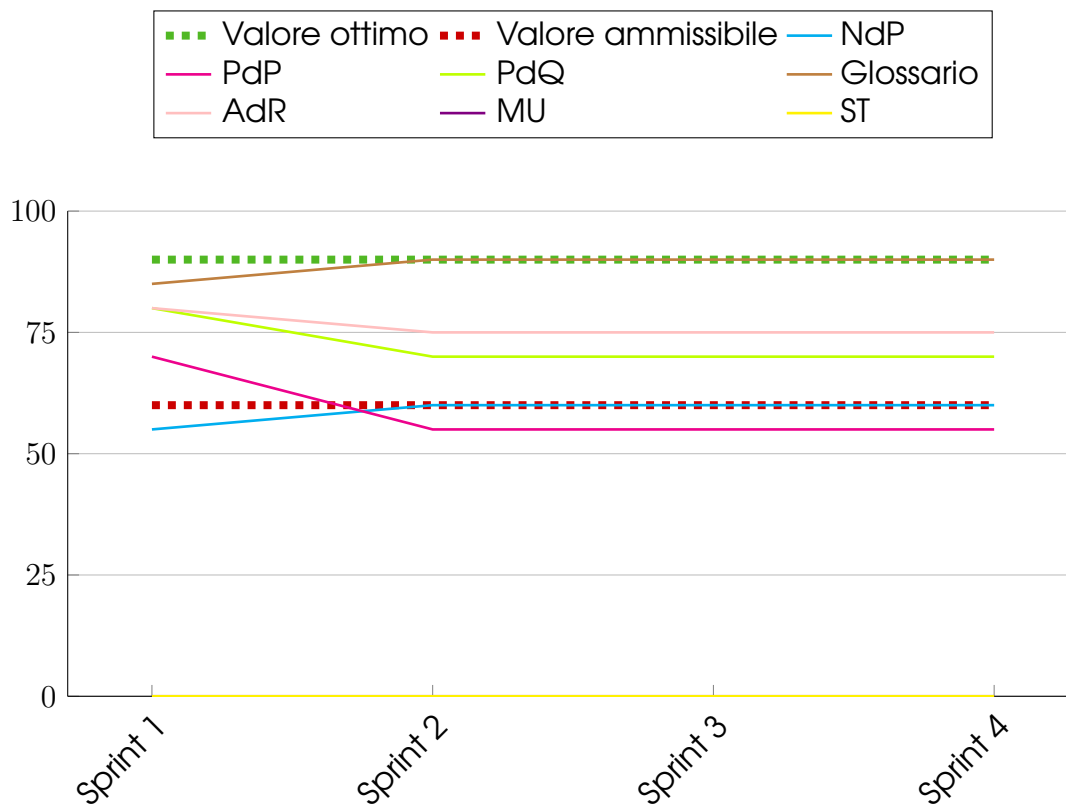


Figure 8: Andamento indice di Gulpease per ciascun documento



#### 4.3.2 23M-CO - Correttezza Ortografica

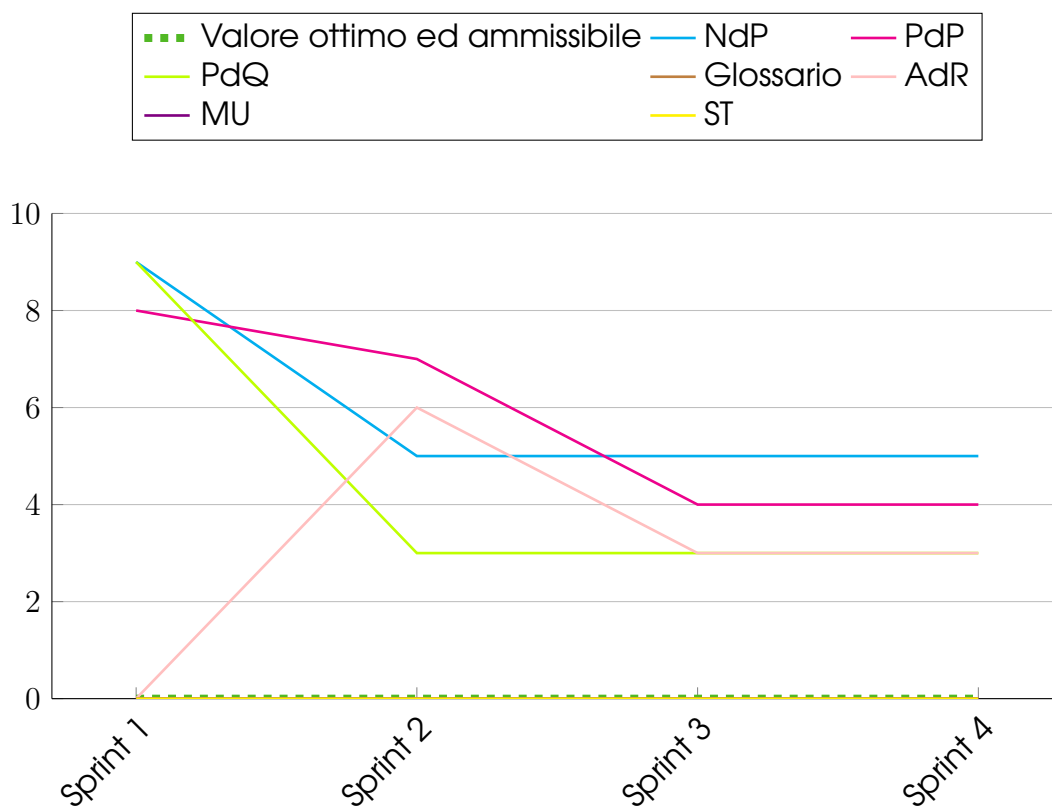


Figure 9: Errori ortografici per ciascun documento



## 4.4 Qualità del processo di Verifica

### 4.4.1 24M-CC - Code coverage

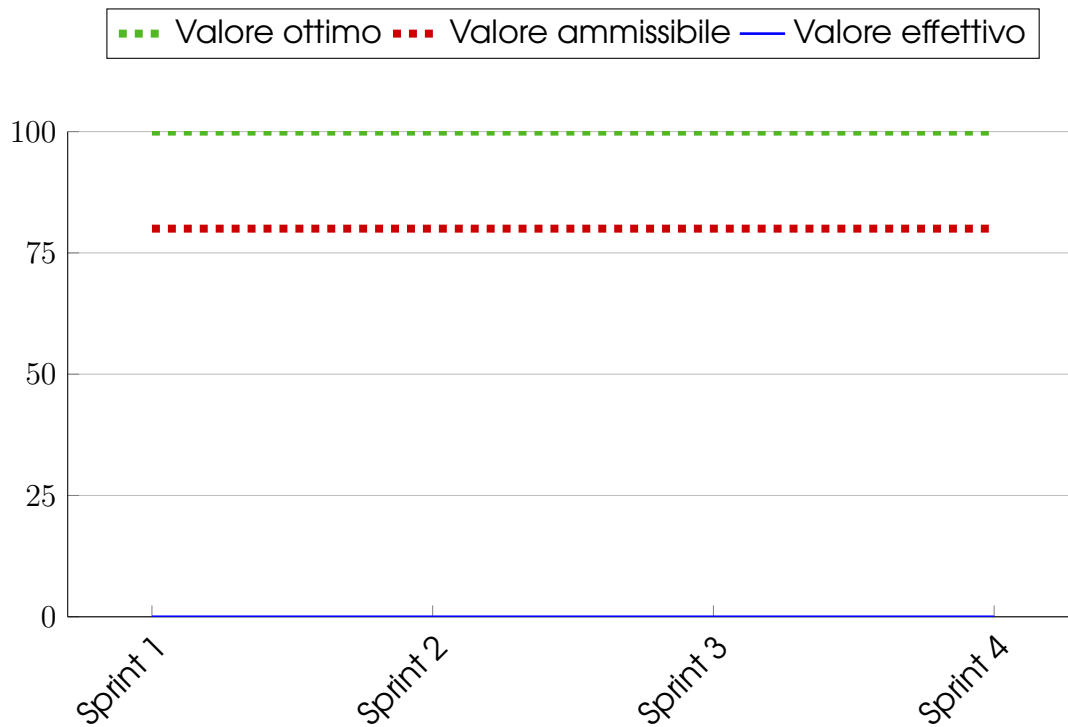


Figure 10: Percentuale di code coverage dei test implementati



#### 4.4.2 25M-BC - Branch coverage

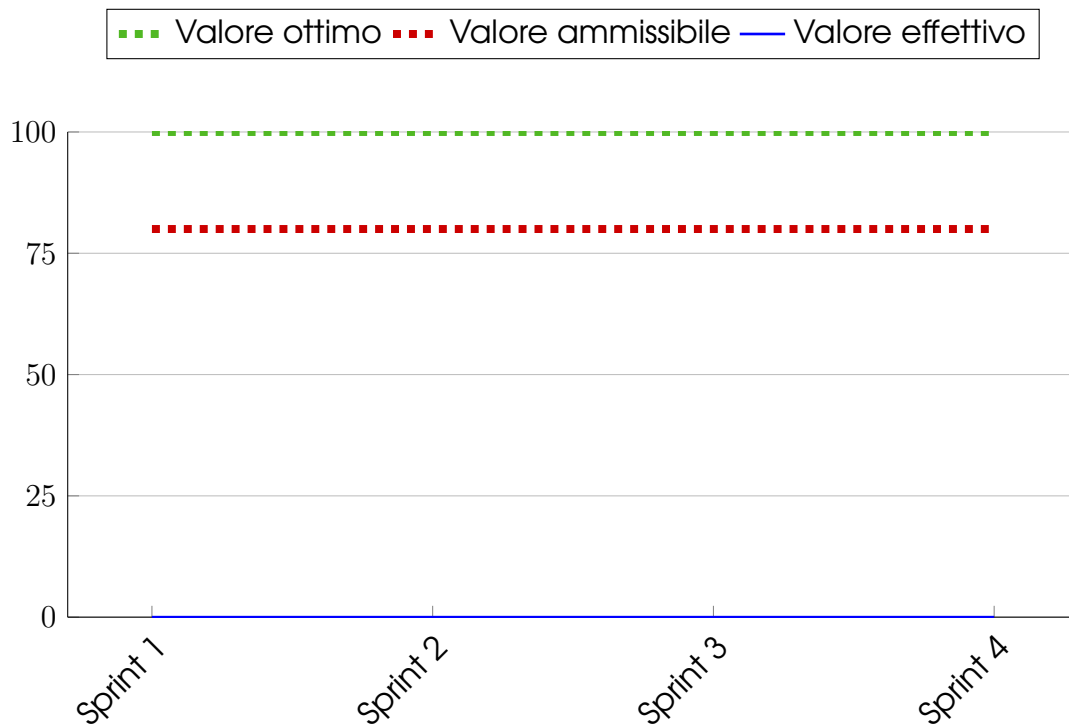


Figure 11: Percentuale di branch coverage dei test implementati





#### 4.4.3 26M-SC - Statement coverage

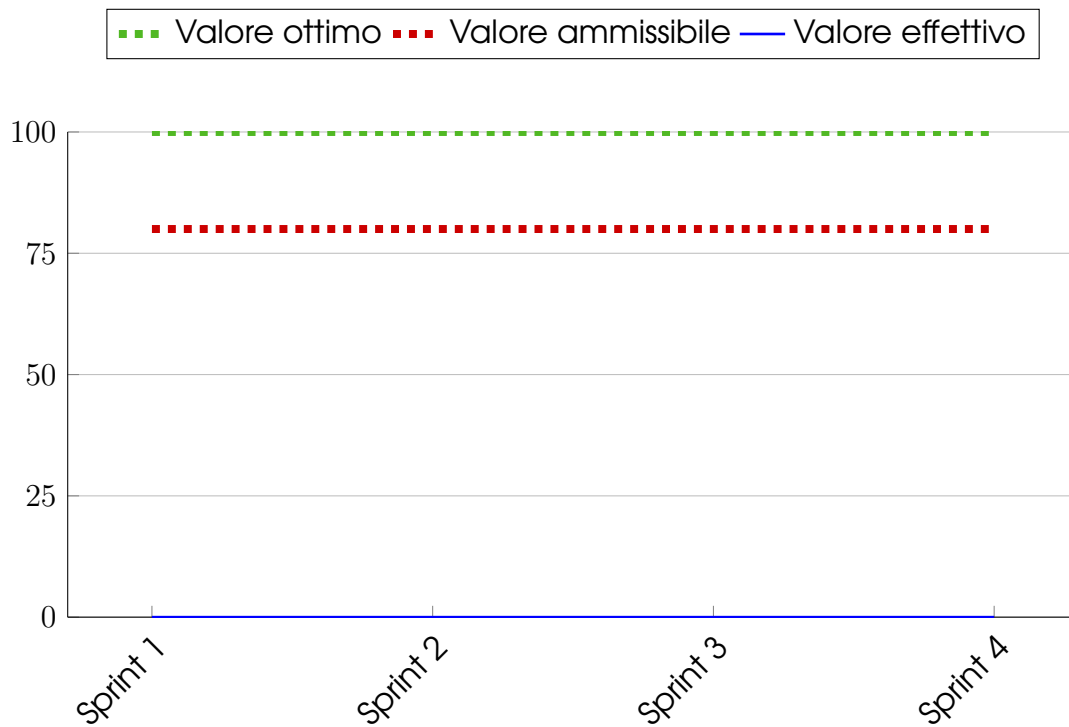


Figure 12: Percentuale di statement coverage dei test implementati



#### 4.4.4 27M-FD - Failure density

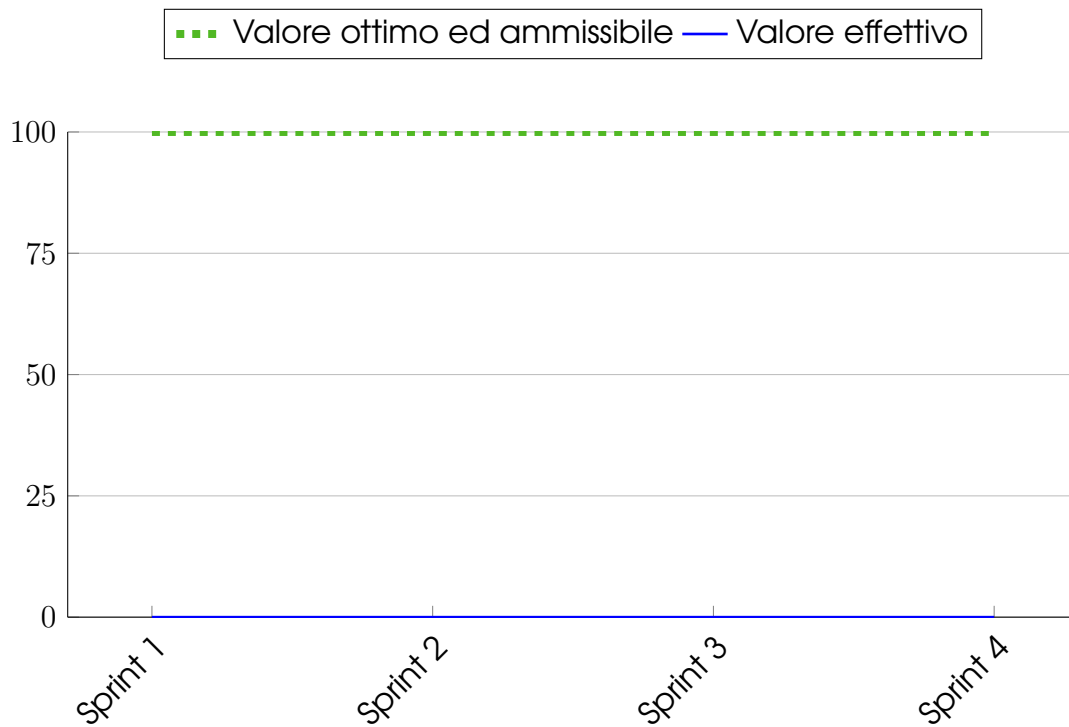


Figure 13: Percentuale di failure density



#### 4.4.5 28M-PTCP - Passed Test Cases Percentage

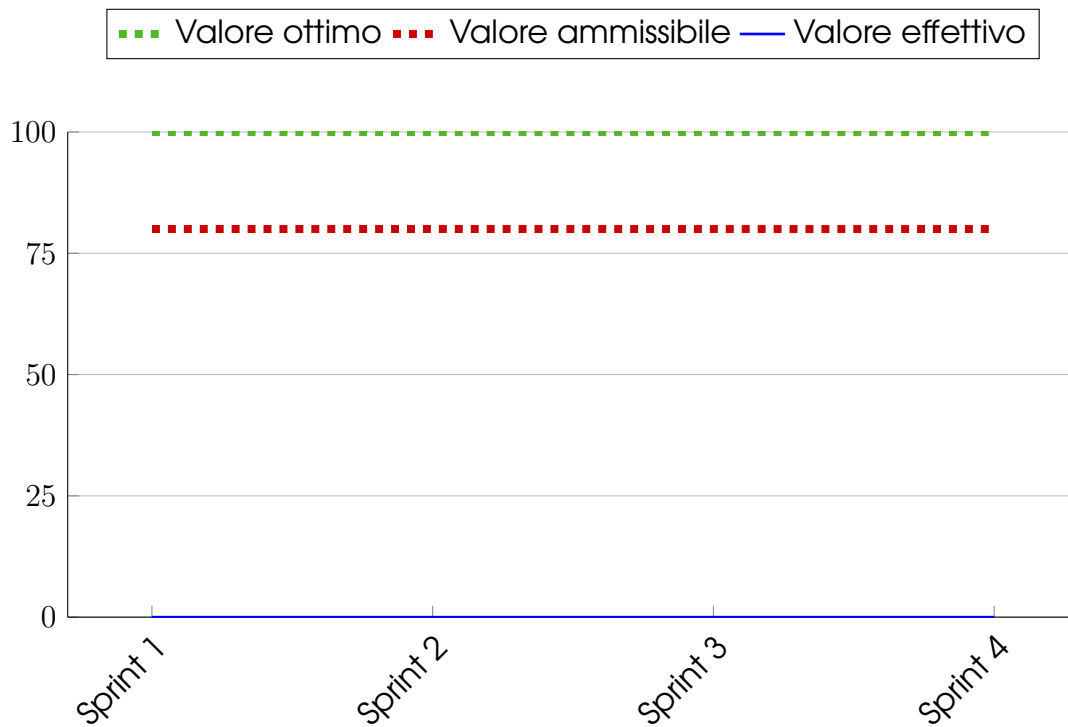


Figure 14: Percentuale di casi di test superati



## 4.5 Qualità del processo di Gestione dei rischi

### 4.5.1 29M-NCR - Non-Calculated Risk

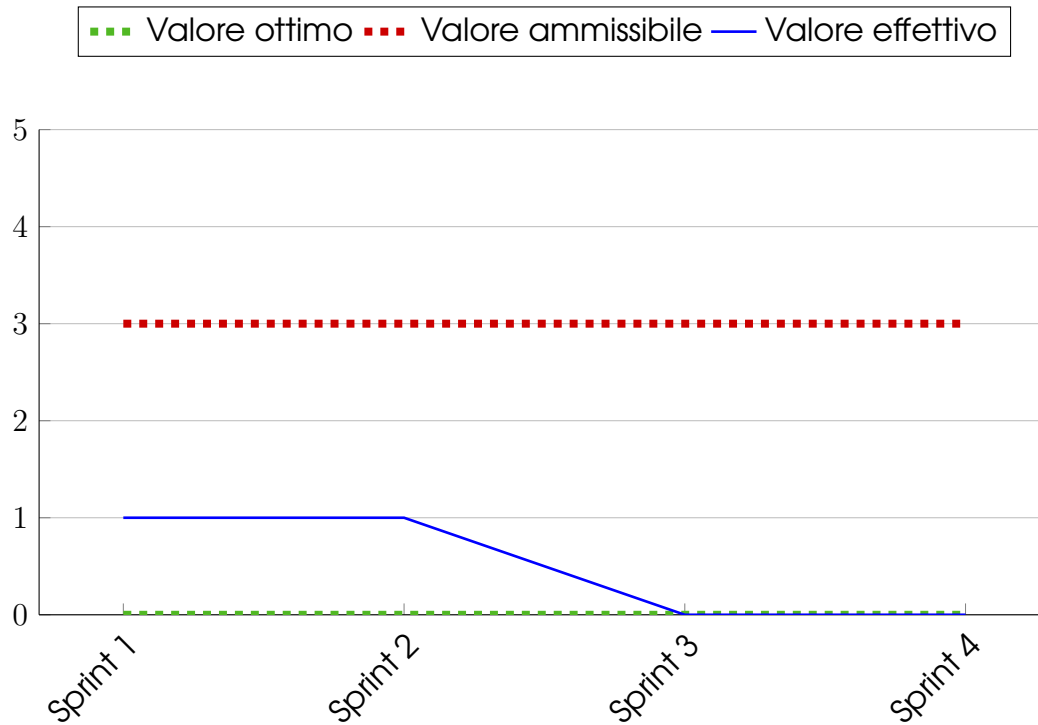


Figure 15: Rischi non calcolati occorsi durante il progetto



## 4.6 Qualità del processo di Gestione della qualità

### 4.6.1 30M-QMS - Quality Metrics Satisfied

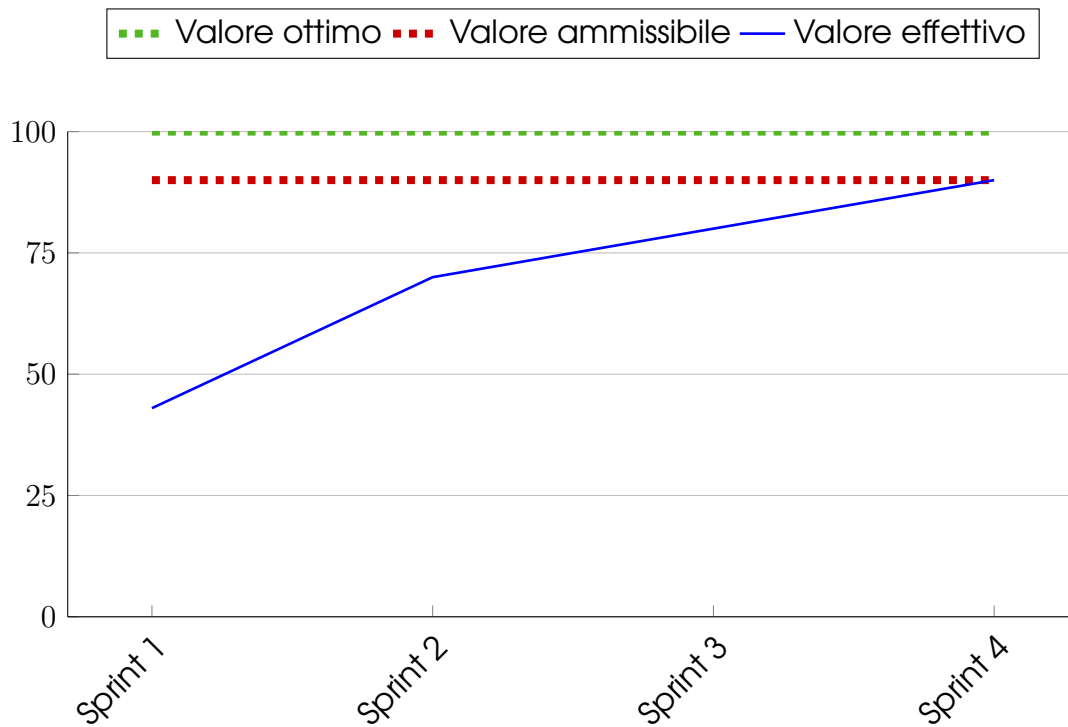


Figure 16: Percentuale di metriche di qualità soddisfatte



#### 4.6.2 31M-TE - Efficienza Temporale

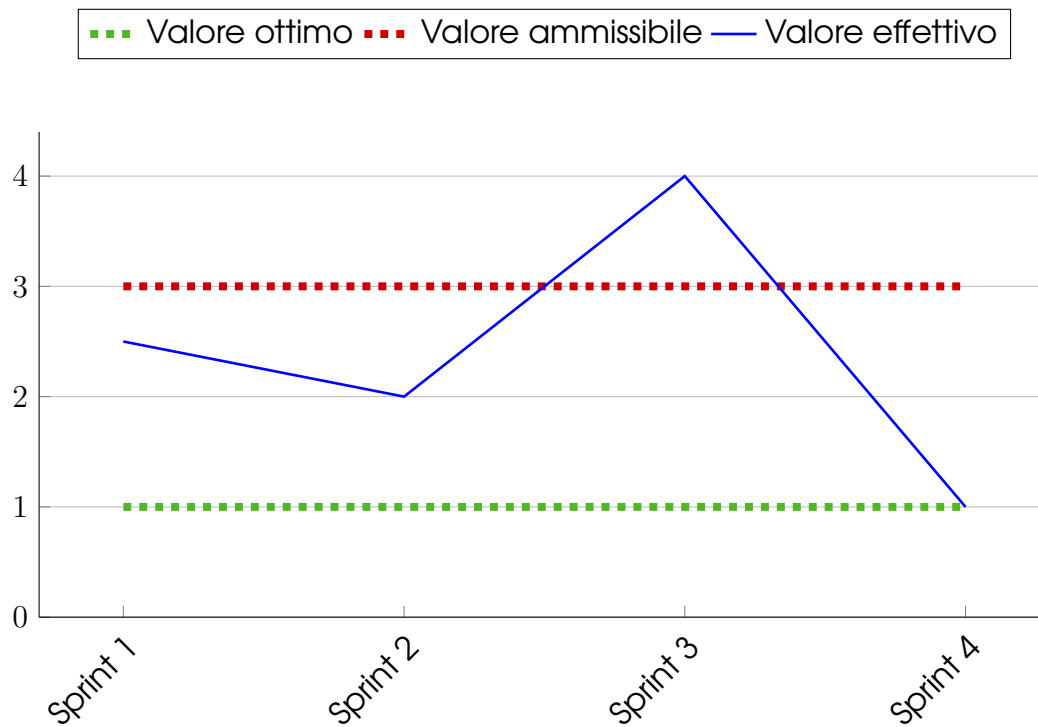


Figure 17: Andamento dell'efficienza temporale



## 4.7 Qualità del processo di Pianificazione

### 4.7.1 32M-RSI - Requirements stability index (RSI)

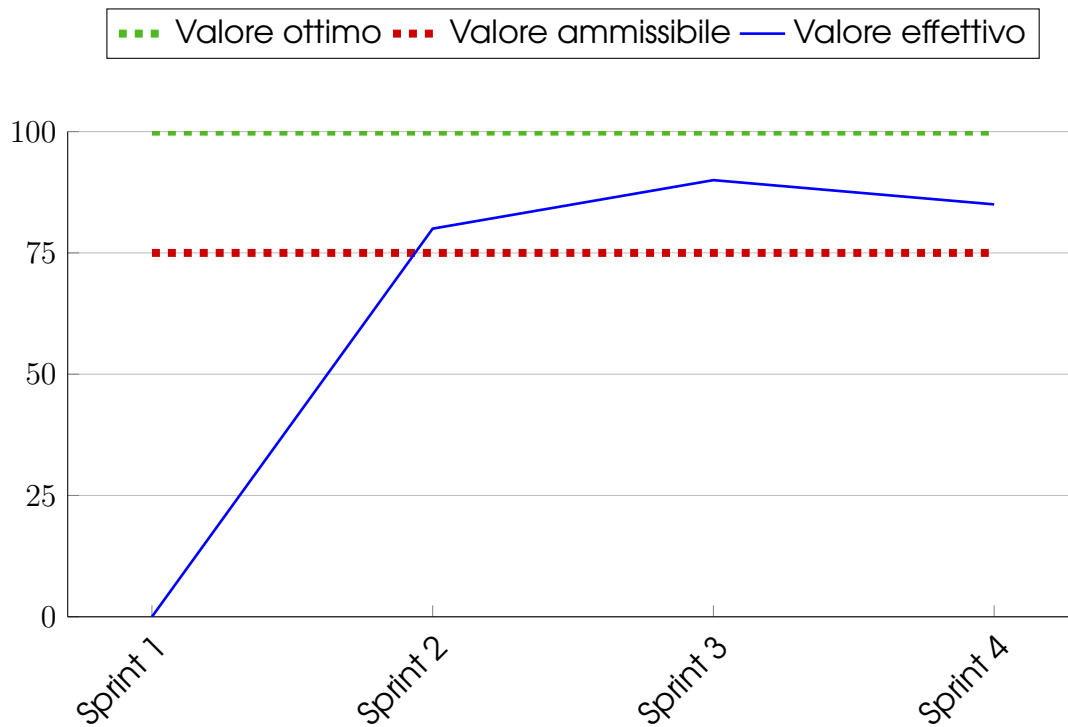


Figure 18: Percentuale di stabilità dei requisiti



## 5 Iniziative di automiglioramento per la qualità

### 5.1 Introduzione

In questa sezione verranno riportate le iniziative di automiglioramento che il nostro gruppo ha deciso di adottare per aumentare la qualità del prodotto e dei processi. Queste iniziative sono state individuate grazie all'esperienza acquisita durante lo svolgimento del progetto e grazie alle valutazioni effettuate sulle attività svolte.

Trattandosi per tutti noi della prima esperienza con un progetto di questa portata, è stato necessario un grande numero di tentativi per comprendere al meglio come organizzarci e come svolgere le attività. Questo ci ha permesso di capire quali sono stati i punti di forza e i punti deboli del nostro lavoro e di individuare le aree in cui è possibile migliorare. Per ciascuna delle difficoltà riscontrate verranno indicate:

- fase del progetto in cui si è verificato il problema;
- descrizione del problema;
- contromisura adottata per risolvere il problema evidenziato.

### 5.2 Problemi rilevati ed iniziative adottate

- **Organizzazione delle riunioni**

- **Fase del progetto:** iniziale;
- **Descrizione:** nelle prime settimane di lavoro, a partire dalla formazione dei gruppi sino ai primi Diari di bordo, si è riscontrata una certa difficoltà nell'organizzazione delle riunioni causata dai vari impegni di ciascun membro (lezioni diverse in orari diversi, lavoro per alcuni, impegni personali) e soprattutto alimentata dalle diverse riunioni che si accumulavano (SAL con l'azienda prima e Diari di bordo poi) portando a una certa confusione e a un rallentamento delle attività;
- **Contromisura:** abbiamo deciso di effettuare le riunioni a distanza tramite la piattaforma *Discord* e di fissare un giorno e un orario durante la settimana per ciascuna tipologia di incontro in maniera tale da rispettare le disponibilità di ogni membro; qualora qualcuno, per impegni di natura eccezionale, non abbia modo di essere presente potrà successivamente informarsi sui contenuti trattati attraverso i verbali che verranno redatti e messi a disposizione di tutti.





- **Suddivisione compiti**

- **Fase del progetto:** iniziale;
- **Descrizione:** all'inizio del progetto si è riscontrata una certa difficoltà nella suddivisione dei compiti a causa della mancanza di esperienza e della poca conoscenza delle competenze possedute da ciascuno. È risultato dunque difficile il bilanciamento delle mansioni e si sono verificati più volte casi in cui alcuni membri sono stati in grado di completare le attività a loro assegnate in anticipo, e casi opposti in cui il lavoro da svolgere è risultato eccessivo e difficilmente completabile entro i tempi prestabiliti;
- **Contromisura:** abbiamo quindi deciso, come suggerito anche dal professor Vardanega al primo Diario di bordo, di non assegnare preventivamente tutti i compiti da svolgere a ciascun membro, ma piuttosto di metterli in un contenitore condiviso (abbiamo deciso di usare le annotazioni di *ClickUp*) e di permettere a ciascun membro di prendere in autonomia i compiti da svolgere, così che chiunque finisca in anticipo possa prenderne altri; in questo modo siamo riusciti a svolgere le attività in modo più equo e a completare i compiti entro i tempi prestabiliti.

- **Familiarità con le tecnologie**

- **Fase del progetto:** intermedia;
- **Descrizione:** durante lo svolgimento del progetto ci siamo resi conto che la mancanza di familiarità con le tecnologie utilizzate (in particolare con *Docker*, *Grafana* e *Clickhouse*) ha rallentato inizialmente l'attività di sviluppo e ha portato a un aumento del carico di lavoro per alcuni membri del gruppo;
- **Contromisura:** abbiamo deciso di organizzare un incontro di formazione in cui i membri più esperti hanno spiegato ai meno esperti il funzionamento di *Docker* e le modalità di utilizzo. Inoltre, abbiamo deciso di utilizzare la funzionalità di *pair programming* per permettere ai membri meno esperti di lavorare a stretto contatto con quelli più esperti e di apprendere da loro.

## 5.3 Considerazioni finali

Fin da subito il nostro gruppo si è posto come obiettivo principale quello di dotarsi di un *Way of Working* preciso e ben definito, di pianificare ogni singola attività e di prevedere



tutte le possibili difficoltà incontrabili durante lo svolgimento del progetto. Questo per cercare di prevenire i problemi e di fornire delle contromisure efficaci per affrontarli. Inizialmente si sono presentate delle difficoltà dovute all'inesperienza del gruppo in ambito organizzativo. Tuttavia, grazie alla familiarizzazione ottenuta tramite lo svolgimento del progetto e grazie ai consigli e suggerimenti che ci sono stati forniti dai professori e dall'azienda proponente, siamo riusciti a individuare i problemi e a mettere in atto delle contromisure per risolverli.

Questo ci ha permesso di migliorare notevolmente la qualità del nostro lavoro e di svolgere le attività in modo più efficiente e più equo. Nonostante ciò siamo anche consapevoli che ci sono ancora molti aspetti su cui possiamo progredire e che ci sono ancora molte iniziative di automiglioramento che possiamo adottare. Siamo convinti che, se continueremo a lavorare con lo stesso impegno e la stessa determinazione che abbiamo dimostrato finora, saremo in grado di ottenere risultati di qualità superiore.