



Piano di Qualifica

Versione 2.0.0

Arena Ivan Antonino
Baesso Nicola
Bousapnameme Ruth Genevieve
Calabrese Luca

Garon Martina
Liva Noemi
Marchiante Marco

Progetto Ingegneria del Software

Dipartimento di Matematica
Università degli Studi di Padova



13 giugno 2023

Contatti: dotseventeam@gmail.com

Registro delle versioni

Versione	Data	Autore	Ruolo	Motivazione
2.0.0	13/06/2023	Ruth Genevieve	Responsabile	Validazione
1.0.8	04/06/2023	Nicola Baesso	Analista	Modifica \$5 (CAP-208). Verificato da Noemi Liva
1.0.7	04/06/2023	Ivan Antonino Arena	Analista	Modifica \$4 (CAP-209). Verificato da Marco Marchiante
1.0.6	04/06/2023	Ivan Antonino Arena	Analista	Modifica \$4 (CAP-235). Verificato da Martina Garon
1.0.5	03/06/2023	Nicola Baesso	Analista	Modifica \$5 e \$6 (CAP-207,CAP-210). Verificato da Martina Garon
1.0.4	12/05/2023	Ruth Genevieve	Analista	Modifica \$4 (CAP-161). Verificato da Ivan Antonino Arena
1.0.3	12/05/2023	Noemi Liva	Analista	Modifica \$5 (CAP-197). Verificato da Nicola Baesso
1.0.2	12/04/2023	Martina Garon	Analista	Modifica \$5 (CAP-144). Verificato da Luca Calabrese
1.0.1	04/04/2023	Nicola Baesso	Analista	Modifica \$2.1 (CAP-135). Verificato da Marco Marchiante
1.0.0	14/02/2023	Martina Garon	Responsabile	Validazione
0.2.0	13/02/2023	Nicola Baesso	Verificatore	Verifica punti modificati
0.1.6	12/02/2023	Noemi Liva	Analista	Stesura \$4.3
0.1.5	12/02/2023	Noemi Liva	Analista	Stesura \$5 e\$6
0.1.4	17/01/2023	Noemi Liva	Analista	Modificato \$2 , \$3 e \$4.0
0.1.3	21/12/2022	Ivan Antonino Arena	Analista	Stesura sezione \$3.4.3.2
0.1.2	19/12/2022	Ruth Genevieve	Analista	Stesura sezioni \$2 e \$3 \$4
0.1.1	10/12/2022	Nicola Baesso	Analista	Modifiche per l'uniformità con altri documenti
0.1.0	10/12/2022	Noemi Liva	Verificatore	Verifica generale
0.0.3	09/12/2022	Nicola Baesso	Analista	Stesura \$2
0.0.2	09/12/2022	Nicola Baesso	Analista	Stesura \$1
0.0.1	09/12/2022	Nicola Baesso	Analista	Stesura documento

Tabella 1: Registro di versionamento del documento

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Maturità del documento	1
1.5	Riferimenti	1
1.5.1	Riferimenti normativi	1
1.5.2	Riferimenti informativi	1
2	Qualità di processo	2
2.1	Introduzione	2
2.2	Processi Primari	2
2.2.1	Obiettivi di qualità	2
2.2.2	Metriche di qualità	2
2.2.2.1	<i>Budget cost of work scheduled (BCS)</i>	2
2.2.2.2	<i>Actual cost of work performed (ACS)</i>	2
2.2.2.3	<i>Scheduled variance</i>	3
2.2.2.4	<i>Budget variance</i>	3
2.2.2.5	<i>Bugs for Line of Code</i>	3
2.3	Processi di supporto	3
2.3.1	Obiettivi di qualità	3
2.3.2	Metriche di qualità	4
2.3.2.1	Indice Gulpease	4
2.3.2.2	<i>Quality Metrics Satisfied</i>	4
2.3.2.3	<i>Code coverage</i>	4
2.3.2.4	<i>Passed test cases percentage</i>	4
3	Qualità di prodotto	5
3.1	Funzionalità	5
3.1.1	Obiettivi di qualità	5
3.1.2	Metriche di qualità	5
3.1.2.1	<i>Requirement Coverage</i>	5
3.2	Affidabilità	6
3.2.1	Obiettivi di qualità	6
3.2.2	Metriche di qualità	6
3.2.2.1	<i>Code Coverage</i>	6
3.2.2.2	<i>Branch Coverage</i>	6
3.2.2.3	<i>Failure Density</i>	7
3.2.2.4	Presenza di <i>bug</i>	7
3.2.2.5	Successo dei test	7
3.3	Usabilità	7
3.3.1	Obiettivi di qualità	7
3.3.2	Metriche di qualità	7
3.3.2.1	Facilità di utilizzo	7
3.4	Manutenibilità	8
3.4.1	Obiettivi di qualità	8
3.4.2	Metriche di qualità	8
3.4.3	Profondità della gerarchia	8
3.4.3.1	Complessità ciclomatica	8
3.4.3.2	<i>Code Smell</i>	9

3.5	Efficienza	9
3.5.1	Obiettivi di qualità	9
3.5.2	Metriche di qualità	9
3.5.2.1	Uso Risorse	9
4	Specifica dei test	10
4.1	Test di unità	11
4.2	Test di regressione	13
4.3	Test di integrazione	13
4.4	Test di sistema	14
4.5	Test di accettazione	15
5	Resoconto delle attività di verifica	16
5.1	Processi	16
5.1.1	Fornitura	16
5.1.1.1	MPC01 - <i>Budget cost of work scheduled</i> (BCS)	16
5.1.1.2	MPC02 - <i>Actual cost of work performed</i> (ACS)	17
5.1.1.3	MPC03 - <i>Scheduled variance</i>	17
5.1.1.4	MPC04 - <i>Budget variance</i>	18
5.1.1.5	MPC05 - <i>Bugs for Line of Code</i>	19
5.1.2	Sviluppo	19
5.1.2.1	MPC06 - <i>Indice Gulpease</i>	19
5.1.2.2	MPC07 - <i>Quality Metrics Satisfied</i>	20
5.1.2.3	MPC08 - <i>Code Coverage</i>	20
5.1.2.4	MPC09 - <i>Passed test cases percentage</i>	21
5.1.3	Qualità di prodotto	22
5.1.3.1	MPD01 - <i>Requirement Coverage</i>	22
5.1.3.2	MPD02 - <i>Code Coverage</i>	22
5.1.3.3	MPD03 - <i>Branch coverage</i>	23
5.1.3.4	MPD04 - <i>Failure Density</i>	23
5.1.3.5	MPD05 - <i>Presenza di bug</i>	24
5.1.3.6	MPD06 - <i>Successo dei test</i>	24
5.1.3.7	MPD07 - <i>Facilità di utilizzo</i>	25
5.1.3.8	MPD08 - <i>Profondità della gerarchia</i>	25
5.1.3.9	MPD09 - <i>Complessità ciclomatica</i>	26
5.1.3.10	MPD10 - <i>Code smell</i>	26
5.1.3.11	MPD11 - <i>Uso risorse</i>	27
5.1.4	Test	28
5.1.4.1	Test di unità	28
5.1.4.2	Test di integrazione	28
5.1.4.3	Test di sistema	29
5.1.4.4	Test di accettazione	29
6	Valutazioni per il miglioramento	30
6.1	Organizzazione	30
6.1.1	Incontri di gruppo	30
6.1.2	Organizzazione delle attività	30
6.1.3	Comunicazione all'interno del gruppo	30
6.2	Ruoli	30
6.2.1	Analista	30
6.2.2	Progettista	30
6.3	Strumenti di lavoro	31

6.3.1	Poca conoscenza generale delle tecnologie scelte	31
6.3.2	Jira	31
6.4	Prodotto	31
6.4.1	Funzionalità	31
6.4.2	Usabilità	31
6.4.3	Prestazioni	31
6.4.4	Sicurezza	31
6.4.5	Compatibilità	32

Elenco delle figure

1	<i>Schedule variance</i>	17
2	<i>Budget variance</i>	18
3	Bugs per linee di codice	19
4	Indice di Gulpease	19
5	Quality Metrics Satisfied	20
6	Code coverage	20
7	Percentuale di test cases passati	21
8	Requirements Coverage	22
9	Code Coverage	22
10	Branch coverage	23
11	Failure Density	23
12	Presenza di bugs	24
13	Successo dei test	24
14	Facilità di utilizzo	25
15	Profondità della gerarchia	25
16	Complessità ciclomatica	26
17	Code smell	26
18	Uso risorse	27
19	Risultati test di unità	28
20	Risultati test di integrazione	28
21	Risultati test di sistema	29
22	Risultati test di accettazione	29

Elenco delle tabelle

1	Registro di versionamento del documento	1
2	Obiettivi di qualità dei processi primari	2
3	Metriche di qualità dei processi primari	2
4	Obiettivi di qualità dei processi di supporto	3
5	Metriche di qualità dei processi di supporto	4
6	Obiettivi di qualità per la funzionalità	5
7	Metriche di qualità per la funzionalità	5
8	Obiettivi di qualità per l'affidabilità	6
9	Metriche di qualità per l'affidabilità	6
10	Obiettivi di qualità per l'usabilità	7
11	Metriche di qualità per l'usabilità	7
12	Obiettivi di qualità per la manutenibilità	8
13	Metriche di qualità per la manutenibilità	8
14	Obiettivi di qualità per l'efficienza	9

15	Metriche di qualità per l'efficienza	9
16	Test di unità	11
17	Test di unità	12
18	Test di integrazione	13
19	Test di sistema	14
20	Test di sistema	15
21	Test di accettazione 1	15
22	Test di accettazione 2	16
23	MPC01 - <i>Budget cost of work scheduled</i> (BCS)	16
24	MPC02 - <i>Actual cost of work performed</i> (ACS)	17

1 Introduzione

1.1 Scopo del documento

In questo documento si illustrano i metodi di **verifica** e **validazione** adottati dal gruppo .7 al fine di garantire la qualità di prodotto e di processo. Saranno inoltre riportati i risultati delle verifiche effettuate sul prodotto, al fine di correggere nel minor tempo possibile eventuali problemi rilevati, minimizzando lo spreco delle risorse.

1.2 Scopo del prodotto

Data la maggiore influenza di intelligenze artificiali sempre più complesse e sistemi informatici robotizzati, è importante sviluppare dei metodi che permettano di distinguere se la persona che sta interagendo con un sistema sia effettivamente una persona fisica o dimostri i comportamenti di uno strumento automatico. L'obiettivo del Team **Dot Seven** e dell'azienda **Zucchetti s.p.a** è quindi quello di creare un sistema CAPTCHA (*Completely Automated Public Test to tell Computers and Humans Apart*) in grado di distinguere le macchine dall'umano.

1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzate è stato creato un documento denominato "Glossario". Questo documento contiene tutti i termini tecnici scelti dal gruppo e utilizzati nei vari documenti con le relative definizioni.

1.4 Maturità del documento

Il presente documento è redatto con un approccio incrementale, per permettere di poter trattare nuove o ricorrenti questioni in modo rapido ed efficiente, sulla base di decisioni concordate tra tutti i membri del gruppo.

1.5 Riferimenti

1.5.1 Riferimenti normativi

- Capitolato d'appalto C1 - CAPTCHA: Umano o Sovrumano?:
<https://www.math.unipd.it/~tullio/IS-1/2022/Progetto/C1.pdf> (data ultimo accesso al link: 13 giugno 2023).

1.5.2 Riferimenti informativi

- Slide T08 del corso di Ingegneria del Software - Qualità di prodotto:
<https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T08.pdf> (data ultimo accesso al link: 13 giugno 2023);
- Slide T09 del corso di Ingegneria del Software - Qualità di processo:
<https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T09.pdf> (data ultimo accesso al link: 13 giugno 2023).

2 Qualità di processo

2.1 Introduzione

Per garantire la qualità di processo, il gruppo ha deciso di adottare lo standard **ISO 9001:2015**¹. Tale norma permette al gruppo di seguire il ciclo di Deming, meglio conosciuto come "PDCA", che prevede un approccio iterativo funzionale per l'attuazione di un miglioramento continuo.

2.2 Processi Primari

2.2.1 Obiettivi di qualità

Processo	Descrizione	Metriche
Fornitura	Processo che ha lo scopo di scegliere le procedure e le risorse necessarie per lo sviluppo del progetto	MPC01, MPC02, MPC03, MPC04
Sviluppo	Processo che comprende le attività e i compiti per realizzare il prodotto software richiesto	MPC05

Tabella 2: Obiettivi di qualità dei processi primari

2.2.2 Metriche di qualità

Metrica	Nome	Valore accettabile	Valore preferibile
MPC01	<i>Budget cost of work scheduled (BCS)</i>	≥ 0	≥ 0
MPC02	<i>Actual cost of work performed (ACS)</i>	\leq budget totale	$ACS \leq BCS$
MPC03	<i>Scheduled variance</i>	$\geq -15\%$	0 %
MPC04	<i>Budget variance</i>	$\geq -10\%$	0 %
MPC05	<i>Bugs for Line of Code</i>	≤ 70	≤ 25

Tabella 3: Metriche di qualità dei processi primari

2.2.2.1 *Budget cost of work scheduled (BCS)*

Quantificazione del valore del lavoro che si dovrebbe aver raggiunto fino al momento del calcolo, ovvero le ore preventivate.

2.2.2.2 *Actual cost of work performed (ACS)*

Costo totale sostenuto per il lavoro effettivo completato fino ad ora, ovvero le ore di lavoro sostenute.

¹ Riferimento: CSQA. Visitato il 4 Aprile 2022.

2.2.2.3 *Scheduled variance*

Differenza tra il tempo effettivo impiegato e il tempo pianificato per arrivare a tale punto. Indica quanto un progetto è in anticipo (percentuale positiva) o in ritardo (percentuale negativa). Ovvero, il calcolo seguente:

$$SC = \frac{BCS - ACS}{BCS}$$

dove:

- BCS: *budget cost of work scheduled*;
- ACS: *actual cost of work performed*.

2.2.2.4 *Budget variance*

Differenza tra il budget a disposizione per il progetto e quello effettivamente utilizzato. Un valore negativo indica che si sta spendendo più di quello che si sta guadagnando.

$$BC = \frac{BP - BE}{BP}$$

dove:

- BP: budget preventivato;
- BE: budget usato effettivamente.

2.2.2.5 *Bugs for Line of Code*

Il numero di righe di codice contenenti *bug* ed errori al proprio interno.

2.3 Processi di supporto

2.3.1 Obiettivi di qualità

Processo	Descrizione	Metriche
Documentazione	Processo che ha lo scopo di definire gli standard e gli strumenti necessari alla stesura di tutti i documenti del progetto	MPC06
Gestione di qualità	Processo che consiste nel garantire gli obiettivi di qualità imposti per il prodotto	MPC07
Verifica	Processo che ha come obiettivo la valutazione della conformità o meno di un prodotto del progetto	MPC08 e MPC09

Tabella 4: Obiettivi di qualità dei processi di supporto

2.3.2 Metriche di qualità

Metrica	Nome	Valore accettabile	Valore preferibile
MPC06	Indice Gulpease	$\geq 40, \leq 100$	$\geq 60, \leq 100$
MPC07	<i>Quality Metrics Satisfied</i>	$\geq 90 \%$	100%
MPC08	<i>Code Coverage</i>	$\geq 80 \%$	100 %
MPC09	<i>Passed test cases percentage</i>	$\geq 90 \%$	100 %

Tabella 5: Metriche di qualità dei processi di supporto

2.3.2.1 Indice Gulpease

Indice di leggibilità del testo tarato sulla lingua italiana. Considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

$$IG = 89 + \frac{300 \times Nf - 10 \times Nl}{Np}$$

Dove:

- Nf : numero di frasi;
- Nl : numero di lettere;
- Np: numero di parole.

2.3.2.2 *Quality Metrics Satisfied*

Descrive la percentuale di metriche di qualità soddisfatte ovvero il rapporto fra le metriche soddisfatte e quelle totali.

2.3.2.3 *Code coverage*

Misura di quante righe / blocchi / archi del codice vengono eseguiti durante l'esecuzione dei test automatici.

2.3.2.4 *Passed test cases percentage*

Misura la percentuale di *test case* passati dal codice.

3 Qualità di prodotto

Facendo riferimento allo standard **ISO/IEC 9126:2001**, è qui riportato un insieme di caratteristiche che il prodotto deve avere per essere considerato di qualità.

In questa sezione sono esposte le metriche utilizzate per valutare se il prodotto finale possieda o meno tali caratteristiche.

3.1 Funzionalità

La funzionalità è la capacità del prodotto software di fornire funzioni che riescano a soddisfare tutti i requisiti presenti nel documento "Analisi dei Requisiti (versione 2.0.0)".

3.1.1 Obiettivi di qualità

Obiettivo	Nome	Descrizione	Metriche
OQP01	Appropriatezza	Il prodotto deve mettere a disposizione un insieme di funzioni conformi agli obiettivi richiesti	MPD01

Tabella 6: Obiettivi di qualità per la funzionalità

3.1.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MPD01	<i>Requirement Coverage</i>	100%	100%

Tabella 7: Metriche di qualità per la funzionalità

3.1.2.1 Requirement Coverage

Rappresenta la copertura dei requisiti definiti dal team mediante l'Analisi dei Requisiti (versione 2.0.0). Tale indice si misura tramite la formula:

$$RC = \frac{N_s}{N_t} \times 100$$

Dove:

- N_s : numero di requisiti obbligatori soddisfatti;
- N_t : numero di requisiti obbligatori totali.

Per il valore ottimale, si prende N_s = numero di requisiti soddisfatti.

3.2 Affidabilità

L'affidabilità è la capacità del prodotto di mantenere prestazioni elevate anche in caso di situazioni anomale o critiche.

3.2.1 Obiettivi di qualità

Obiettivo	Nome	Descrizione	Metriche
OQP02	Maturità	Il prodotto deve evitare che si verifichino errori e malfunzionamenti	MPD02 MPD03
OQP03	Tolleranza agli errori	Il prodotto mantiene alte prestazioni anche in caso di malfunzionamenti o di un uso scorretto	MPD04 MPD05 MPD06

Tabella 8: Obiettivi di qualità per l'affidabilità

3.2.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MPD02	<i>Code coverage</i>	100 %	≥ 80 %
MPD03	<i>Branch coverage</i>	100 %	≥ 80 %
MPD04	<i>Failure Density</i>	0 %	15 %
MPD05	Presenza di <i>bug</i>	≤ 5	≤ 20
MPD06	Successo dei test	100 %	≥ 80 %

Tabella 9: Metriche di qualità per l'affidabilità

3.2.2.1 Code Coverage

Indica la percentuale di codice eseguito durante i test.

Un'alta percentuale di copertura indica che il codice effettivamente testato è elevato, pertanto è più probabile che non contenga *bug* nascosti, rispetto a codice con una percentuale di copertura inferiore.

3.2.2.2 Branch Coverage

Indica la percentuale di esecuzione dei *branch*, che si presentano nel codice quando testato.

Un *branch* è un intero ramo di esecuzione. È compito dei test esplorare ogni possibile ramo di esecuzione, in modo da poterne verificare la correttezza. Più è elevato il *branch coverage*, più rami saranno stati testati.

3.2.2.3 Failure Density

Indica l'affidabilità di un prodotto software. Si ricava dal rapporto tra i test eseguiti sul prodotto ed i test che esso ha fallito.

$$FD = \frac{Tf}{Te} \times 100$$

Dove:

- Tf : numero di test falliti;
- Te : numero di test eseguiti.

3.2.2.4 Presenza di bug

Indica il numero di *bug* presenti nel codice.

3.2.2.5 Successo dei test

Indica la percentuale di test passati con successo.

3.3 Usabilità

Capacità del prodotto di essere di facile comprensione e utilizzo da parte degli utenti.

3.3.1 Obiettivi di qualità

Obiettivo	Nome	Descrizione	Metriche
OQP04	Comprensibilità	L'utente deve essere in grado di comprendere le funzionalità offerte dal prodotto e di utilizzarle	MPD07

Tabella 10: Obiettivi di qualità per l'usabilità

3.3.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MPD07	Facilità di utilizzo	5	≤ 8

Tabella 11: Metriche di qualità per l'usabilità

3.3.2.1 Facilità di utilizzo

La facilità di utilizzo è data dalla velocità con la quale l'utente riesce a reperire le informazioni di cui necessita. Nella fattispecie, viene calcolato il numero di click richiesti per superare il CAPTCHA.

Il valore preferibile è quindi 5, considerando uno per lo *username*, uno per la password, due per le risposte al CAPTCHA e uno per la conferma dei dati. Nel caso di dati errati, saranno richiesti ulteriori 3 click.

3.4 Manutenibilità

Capacità del prodotto di essere mantenuto, includendo correzioni, miglioramenti o adattamenti.

3.4.1 Obiettivi di qualità

Obiettivo	Nome	Descrizione	Metriche
OQP05	Analizzabilità	Facilità con la quale è possibile analizzare il codice per localizzare un errore	MPD08
OQP06	Modificabilità	Capacità del prodotto di permettere l'implementazione di una modifica	MPD09 MPD10

Tabella 12: Obiettivi di qualità per la manutenibilità

3.4.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MPD08	Profondità della gerarchia	≤ 2	≤ 4
MPD09	Complessità ciclomatica	≤ 10	≤ 20
MPD10	<i>Code smell</i>	≤ 10	≤ 50

Tabella 13: Metriche di qualità per la manutenibilità

3.4.3 Profondità della gerarchia

Il numero di livelli di una gerarchia serve per indicarne la profondità. In generale, più una gerarchia è profonda, più è complessa da analizzare. D'altro canto, se ben progettata, potrebbe essere più facile modificarne solo alcune specifiche parti. Per questo motivo, è opportuno trovare il giusto equilibrio, per renderne semplice sia l'analisi che la modifica e quindi più efficiente la manutenibilità.

3.4.3.1 Complessità ciclomatica

Indica la complessità di un programma. La complessità ciclomatica di una sezione del codice sorgente è il numero di percorsi linearmente indipendenti al suo interno.

Inizialmente pari a 1, viene incrementata da *branch*, salti e iterazioni. Dato il grafo G del flusso di esecuzione all'interno dell'unità, la complessità ciclomatica si calcola come:

$$v(G) = e - n + p$$

dove:

- e: indica il numero degli archi del grafo;
- n: indica il numero dei nodi del grafo;
- p: indica il numero delle componenti connesse da ogni arco.

3.4.3.2 Code Smell

Indica il numero di difetti di programmazione riconosciuti nel codice sorgente del prodotto.

I *code smell* rappresentano delle debolezze di progettazione che riducono la qualità del software, a prescindere dall'effettiva correttezza del suo funzionamento.

3.5 Efficienza

La capacità del prodotto software di raggiungere e soddisfare gli obiettivi prefissati con il minor uso di risorse possibile.

3.5.1 Obiettivi di qualità

Obiettivo	Nome	Descrizione	Metriche
OQP07	Uso risorse	Il prodotto deve mettere a disposizione tutti i requisiti e usando le giuste risorse	MPD11

Tabella 14: Obiettivi di qualità per l'efficienza

3.5.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MPD11	Uso risorse	≤ 1	$\leq 1,3$

Tabella 15: Metriche di qualità per l'efficienza

3.5.2.1 Uso Risorse

Si calcola facendo il rapporto fra le risorse utilizzate e quelle a disposizione.

4 Specifica dei test

Vengono qui esposte le strategie scelte per il *testing*, queste avranno la finalità di garantire la correttezza e la qualità del prodotto. I test possono essere di vario tipo, ognuno con lo scopo di individuare difetti software differenti.

Il gruppo ha deciso che, per perseguire la correttezza del prodotto e facilitare la fase di validazione, svolgerà la verifica in parallelo allo sviluppo (modello a V). Nelle tabelle seguenti si utilizzeranno delle sigle:

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

4.1 Test di unità

Vengono effettuati per verificare che il comportamento di ogni singolo componente sia corretto. I test di unità vengono stabiliti nel periodo di progettazione e codifica.

Codice	Descrizione	Stato
TU-1	Si verifica che il componente React App venga renderizzato correttamente.	S
TU-2	Si verifica che il componente React Footer venga renderizzato correttamente.	S
TU-3	Si verifica che i dati inseriti al momento del login siano caricati nel sistema.	S
TU-4	Si verifica che il nome utente inserito non abbia caratteri non accettati.	S
TU-5	Si verifica che la <i>password</i> inserita dall'utente rispetti le linee guida date.	NI
TU-6	Si verifica che l'errore generico sia corretto.	S
TU-7	Si verifica che nel momento della registrazione gli errori non generici siano corretti.	NI
TU-8	Si verifica che in caso di <i>password</i> e nome utente non corretti non venga effettuato il login.	S
TU-9	Si verifica che l'immagine del CAPTCHA venga renderizzata correttamente.	S
TU-10	Si verifica che le risposte del CAPTCHA vengano mostrate all'utente.	S
TU-11	Si verifica che le risposte del CAPTCHA date dall'utente siano inserite nel sistema.	S
TU-12	Si verifica che le immagini del CAPTCHA vengano generate secondo specifica.	S
TU-13	Si verifica che in caso di risposte errate al CAPTCHA, non venga effettuato il login.	S
TU-14	Si verifica che venga generato correttamente un nuovo CAPTCHA se richiesto.	S
TU-15	Si verifica che la creazione di un oggetto test CAPTCHA avvenga correttamente.	S
TU-16	Si verifica che ogni metodo GET delle classi ritorni un oggetto del tipo specificato nel nome del metodo.	S

Tabella 16: Test di unità

Codice	Descrizione	Stato
TU-17	Si verifica che un utente autenticato possa effettuare il <i>logout</i> .	S
TU-18	Si verifica che i <i>link</i> della <i>navbar</i> portino alla pagina corrispondente.	S
TU-19	Si verifica che i campi durante il login siano inizialmente vuoti.	S
TU-20	Si verifica che al momento del test CAPTCHA non ci siano risposte già inserite	S
TU-21	Si verifica che il logo del <i>team</i> venga renderizzato correttamente	S
TU-22	Verificare che il metodo <code>decode(string)</code> della classe JWT ritorna un JSON Web Token.	S
TU-23	Verificare che il metodo <code>checkUser</code> ritorni il valore uno se esiste un utente con il nome e la password specificata dal metodo nella <code>userRepository</code> .	S

Tabella 17: Test di unità

4.2 Test di regressione

I test di regressione hanno lo scopo di verificare che le correzioni o le estensioni effettuate su specifiche unità già testate non danneggino il resto del sistema. Essi consistono nella ripetizione selettiva di test di unità, integrazione e sistema.

4.3 Test di integrazione

Vengono effettuati per verificare che i comportamenti del sistema e delle sue componenti siano corretti, quando queste ultime vengono messe in relazione tra di loro.

Codice	Descrizione	Stato
TI-1	Si verifica che l'integrazione con la libreria di sovrapposizione delle immagini sia gestita correttamente.	S
TI-2	Si verifica che React, utilizzato per il <i>frontend</i> comunichi correttamente con il <i>backend</i> .	S
TI-3	Si verifica che l'integrazione con il <i>database</i> di immagini sia gestita correttamente.	S
TI-4	Si verifica che i dati inseriti durante la registrazione siano caricati correttamente.	NI
TI-5	Si verifica che i dati inseriti durante il login siano caricati correttamente.	S
TI-6	Si verifica che l'applicazione web si interfacci in modo corretto con il sistema CAPTCHA, per una corretta verifica dell'identità cliente	S
TI-7	Si verifica che l'integrazione tra le componenti <i>frontend</i> e <i>backend</i> sia realizzata in modo corretto per la risoluzione e verifica CAPTCHA, gestendo tutte le possibili risposte del <i>backend</i>	S
TI-8	Si verifica che l'interazione tra le pagine del <i>frontend</i> e il <i>database</i> sia gestita in modo corretto durante le fasi di <i>login</i> e registrazione	NI

Tabella 18: Test di integrazione

4.4 Test di sistema

Vengono effettuati per assicurare che i requisiti identificati nella fase dell'analisi dei requisiti siano rispettati.

Codice	Descrizione	Stato	Requisito
TS-1	Si verifica che il sistema CAPTCHA richieda autenticazione per l'utilizzo di alcune delle funzionalità offerte	S	RF01
TS-2	Si verifica che il sistema CAPTCHA riesca ad offrire la possibilità di richiedere un nuovo test CAPTCHA	S	RF02
TS-3	Si verifica che il sistema CAPTCHA riesca a verificare la correttezza di un test CAPTCHA	S	RF03
TS-4	Si verifica che il processo di autenticazione richieda un ID univoco ed un <i>secret</i> condiviso che ne certifichi l'identità	S	RF05
TS-5	Si verifica che, in caso di errori durante l'autenticazione, venga restituito sempre un errore generico, tranne nel caso di parametri vuoti o test CAPTCHA non risolto	S	RF06
TS-6	Si verifica che, nel caso in cui l'ID del CAPTCHA non sia valido, il sistema restituisca lo stesso errore generico di invalidità dell'ID CAPTCHA	S	RF07
TS-7	Si verifica che, nel caso in cui l'ID del CAPTCHA non sia valido, venga richiesto un nuovo CAPTCHA	S	RF07
TS-8	Si verifica che, durante la validazione del risultato del test CAPTCHA, la mancata comunicazione di qualche dato o un'errata formattazione delle risposte vengano segnalate con errore specifici	S	RF08
TS-9	Si verifica che il test CAPTCHA riesca a limitare l'efficacia di un attacco DOS	NI	RF09
TS-10	Si verifica che il test CAPTCHA protegga il sistema dall'accesso di <i>bot-IA</i> nei servizi critici	S	RF10

Tabella 19: Test di sistema

Codice	Descrizione	Stato	Requisito
TS-11	Si verifica che ci sia una pagina di login protetta da CAPTCHA	S	RF11
TS-12	Si verifica che ci sia una pagina di registrazione protetta da CAPTCHA	NI	RF12
TS-13	Si verifica che gli utenti autenticati possano visualizzare un forum con post	NI	RF13
TS-14	Si verifica che gli utenti autenticati possano ordinare i post secondo popolarità, data o attività	NI	RF14
TS-15	Si verifica che gli utenti autenticati possano visualizzare, se presenti, i commenti di un post	NI	RF15
TS-16	Si verifica che gli utenti autenticati possano inserire nuovi post e commentare quelli già presenti	NI	RF16
TS-17	Si verifica che sia possibile disconnettersi dall'app	S	RF18

Tabella 20: Test di sistema

4.5 Test di accettazione

Vengono effettuati insieme al proponente nella fase di collaudo: servono per verificare che il sistema sia conforme alle aspettative e in caso di esito positivo il prodotto viene rilasciato.

Codice	Descrizione	Stato	Requisiti associati
TA-1	Si verifica che l'utente possa inserire correttamente i dati per effettuare il login.	S	RF24
TA-2	Si verifica che l'utente possa risolvere il test CAPTCHA offerto dal sistema.	S	RF03,RF04,RF05,RF06,RF07,RF08,RF10,RF11,RF12,RF13,RF17,RF18,RF19,RF22
TA-3	Si verifica che l'utente visualizzi un messaggio di errore in caso i dati inseriti non siano corretti o siano vuoti.	S	RF27

Tabella 21: Test di accettazione 1

Codice	Descrizione	Stato	Requisiti associati
TA-4	Si verifica che l'utente visualizzi un messaggio di errore relativo all'autenticazione fallita nel caso il test CAPTCHA non sia stato risolto correttamente.	S	RF23
TA-5	Si verifica che l'utente visualizzi un messaggio di errore nel caso il risultato del test CAPTCHA non sia stato inviato correttamente.	S	RF09,RF14,RF16, RF20,RF21
TA-6	Si verifica che l'utente, dopo aver visualizzato un messaggio di errore relativo all'autenticazione fallita, possa subito effettuare un altro test CAPTCHA.	S	RF15
TA-7	Si verifica che l'utente possa effettuare la disconnessione dal sistema.	S	RF28

Tabella 22: Test di accettazione 2

5 Resoconto delle attività di verifica

5.1 Processi

5.1.1 Fornitura

5.1.1.1 MPC01 - *Budget cost of work scheduled* (BCS)

Periodo	Ore preventivate	Costo preventivato (€)
Fase di Analisi dei Requisiti	149	3.055,00
Fase di <i>Proof of Concept</i>	125	2.430,00
Fase di Progettazione di dettaglio e codifica	234	4.835,00
Totale	508	10.320,00

Tabella 23: MPC01 - *Budget cost of work scheduled* (BCS)

5.1.1.2 MPC02 - *Actual cost of work performed* (ACS)

Periodo	Ore effettive	Costo effettivo (€)
Fase di Analisi dei Requisiti	136	3.045,00
Fase di <i>Proof of Concept</i>	106	1.942,50
Fase di Progettazione di dettaglio e codifica	361	7.155,00
Totale	603	12.142,50

Tabella 24: MPC02 - *Actual cost of work performed* (ACS)

5.1.1.3 MPC03 - *Scheduled variance*

I valori sono stati calcolati alla fine delle fasi riportate nel piano di progetto, ovvero la fase di Analisi, del *Proof of Concept* e della progettazione di dettaglio e codifica.

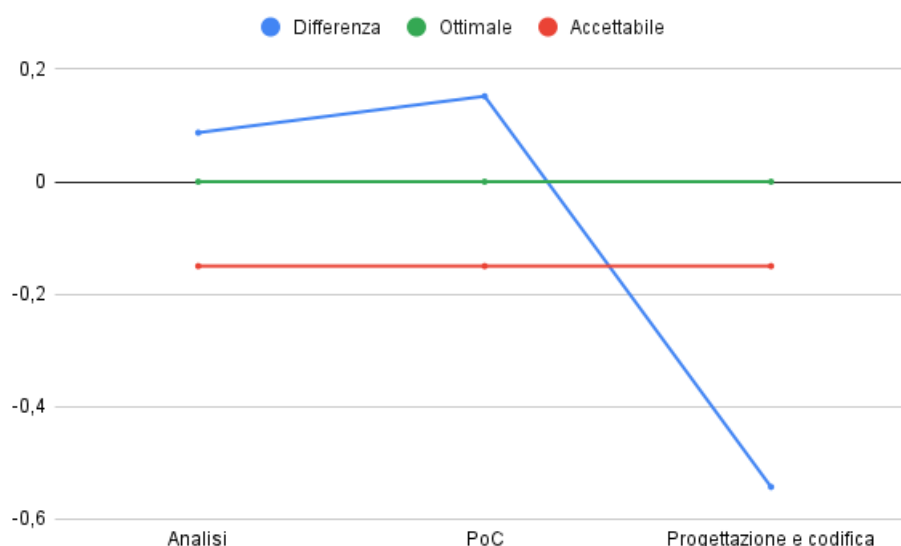


Figura 1: *Schedule variance*

Riguardo le ore preventivate, le attività del gruppo erano inizialmente in positivo, ovvero avevamo usato meno ore di quelle previste, nell'ultima fase sono invece molto in negativo, a causa di ritardi importanti e di un preventivo sbagliato. Nel piano di progetto (versione 3.0.0 capitolo 6.3) viene descritto il motivo di queste ore. Il grafico riguarda la *scheduled variance*.

5.1.1.4 MPC04 - *Budget variance*

I valori sono stati calcolati alla fine delle fasi riportate nel piano di progetto, ovvero la fase di Analisi, del *Proof of Concept* e della progettazione in dettaglio e codifica.

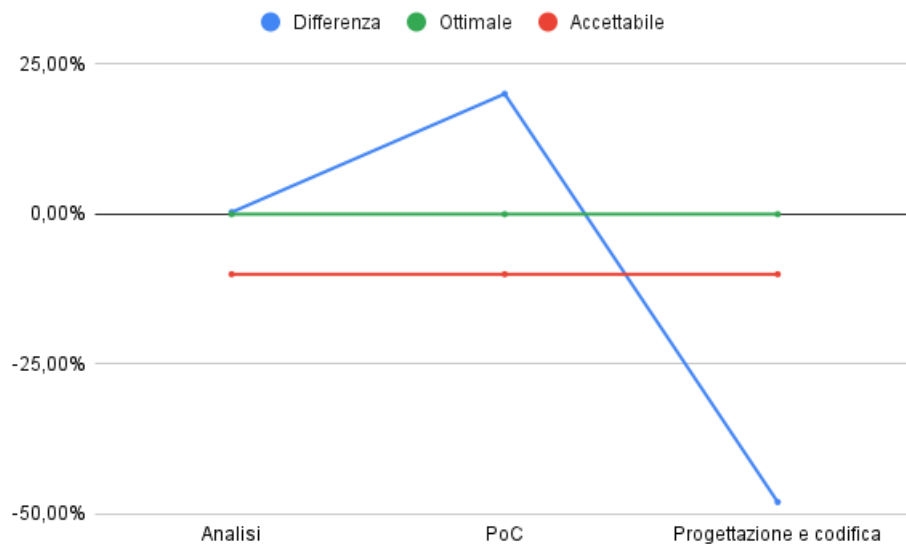


Figura 2: *Budget variance*

Il gruppo ha speso inizialmente meno del previsto a causa di un ritardo sulla tabella di marcia. Ha speso molto più del previsto per la fase di progettazione in dettaglio e codifica, tutti i motivi di queste spese sono descritti nel piano di progetto (versione 3.0.0 capitolo 6.3).

5.1.1.5 MPC05 - Bugs for Line of Code

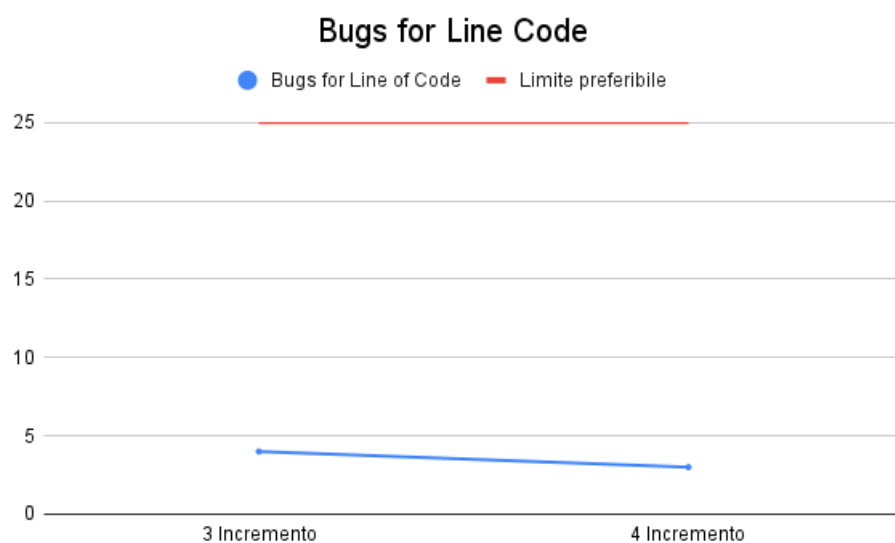


Figura 3: Bugs per linee di codice

5.1.2 Sviluppo

5.1.2.1 MPC06 - Indice Gulpease

È stato calcolato l'indice di Gulpease di ogni documento redatto, escludendo intestazione, registro delle modifiche e dati presenti nelle tabelle, al fine di evitare risultati inesatti. Come riferimento temporale, si usano i periodi elencati nel piano di progetto.

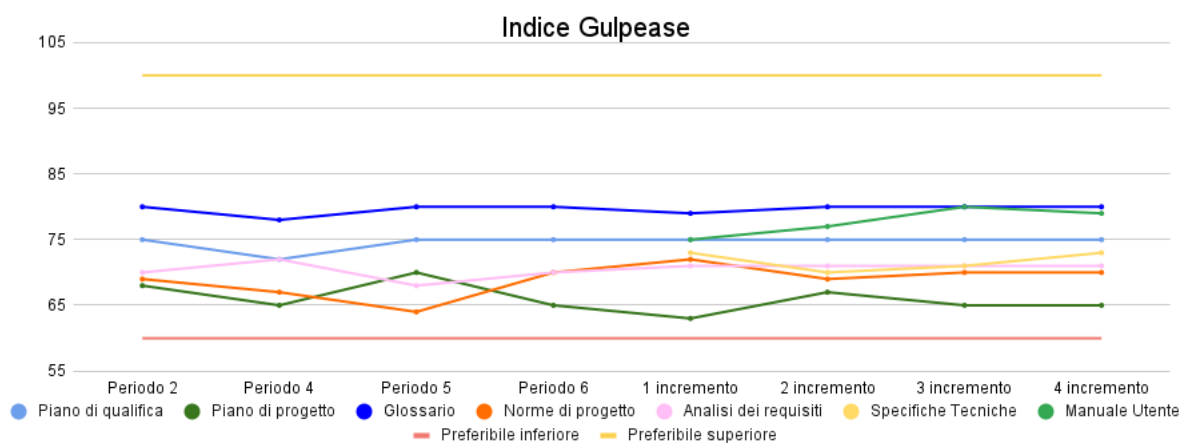


Figura 4: Indice di Gulpease

5.1.2.2 MPC07 - *Quality Metrics Satisfied*

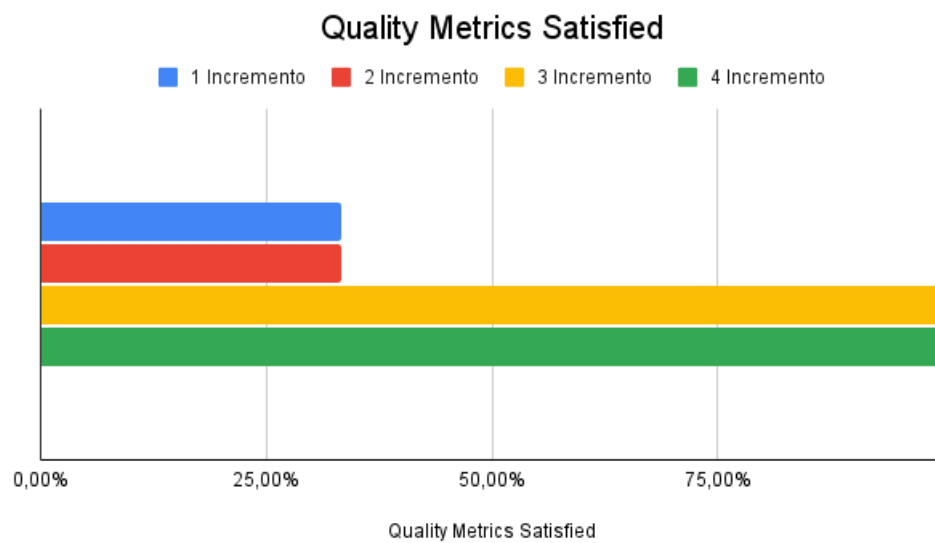


Figura 5: Quality Metrics Satisfied

5.1.2.3 MPC08 - *Code Coverage*

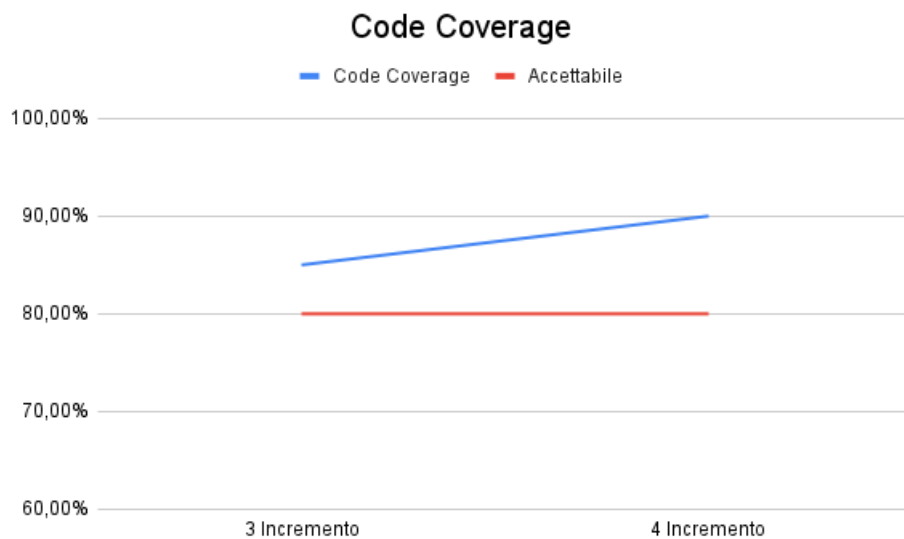


Figura 6: Code coverage

5.1.2.4 MPC09 - *Passed test cases percentage*

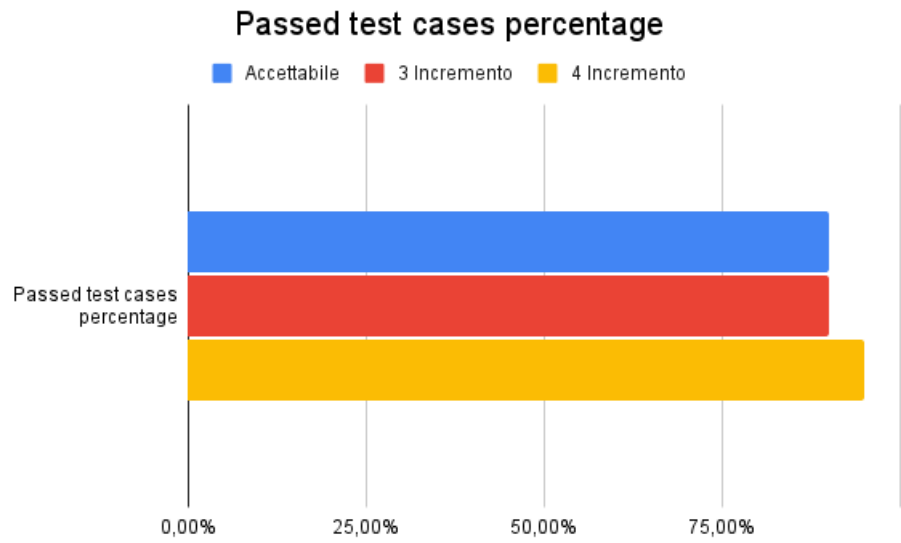


Figura 7: Percentuale di test cases passati

5.1.3 Qualità di prodotto

5.1.3.1 MPD01 - *Requirement Coverage*

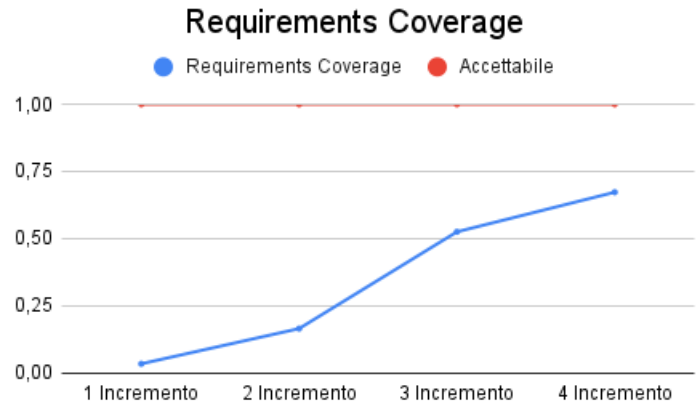


Figura 8: Requirements Coverage

5.1.3.2 MPD02 - *Code Coverage*

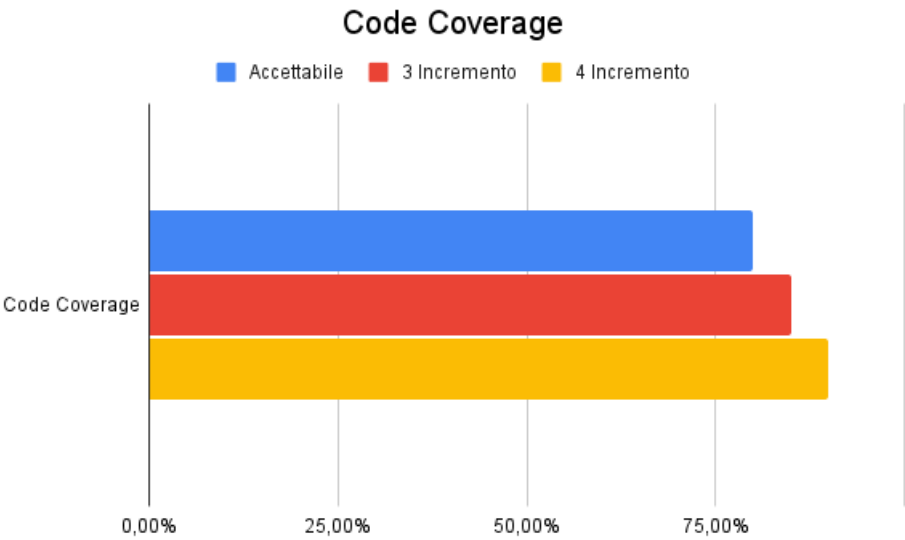


Figura 9: Code Coverage

5.1.3.3 MPD03 - *Branch coverage*

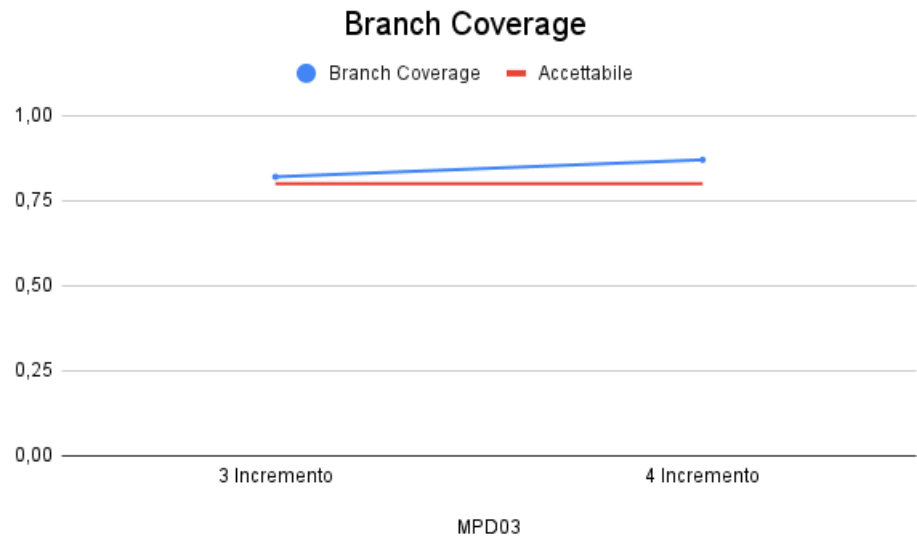


Figura 10: Branch coverage

5.1.3.4 MPD04 - *Failure Density*

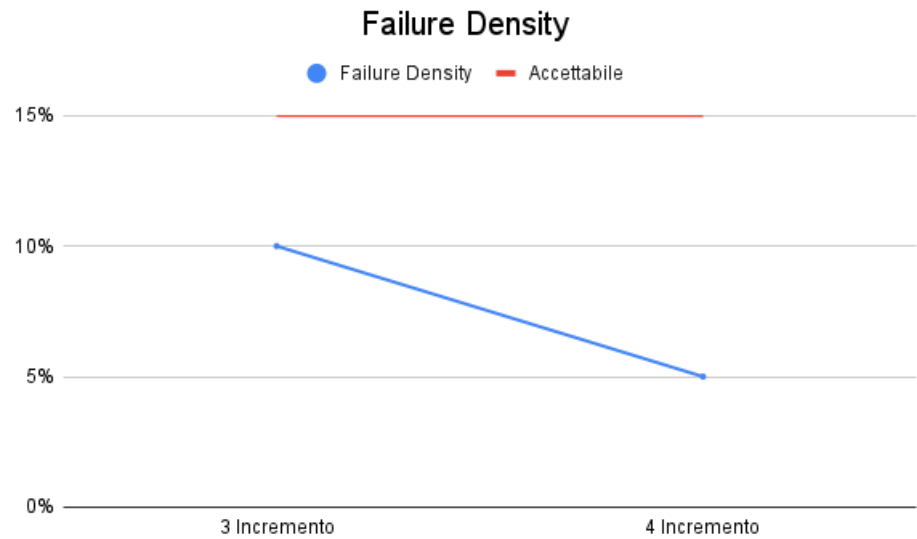


Figura 11: Failure Density

5.1.3.5 MPD05 - Presenza di *bug*

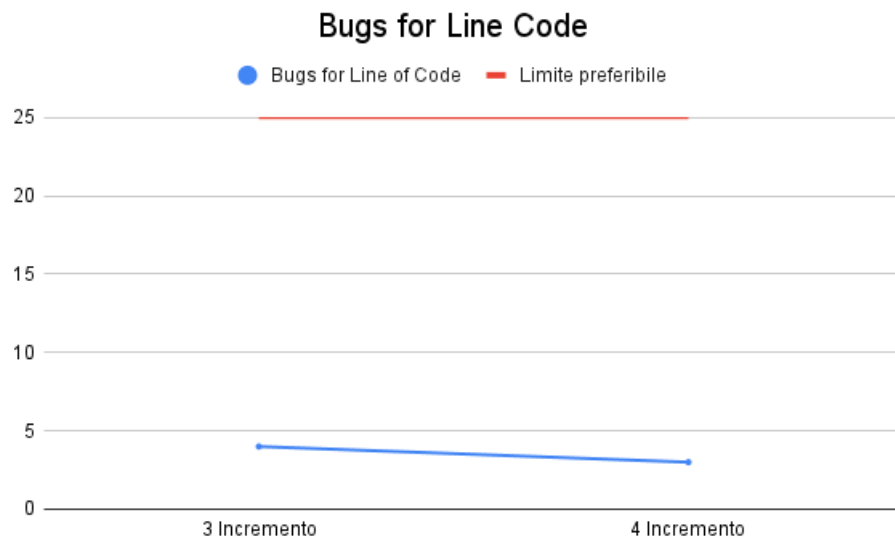


Figura 12: Presenza di bugs

5.1.3.6 MPD06 - Successo dei test



Figura 13: Successo dei test

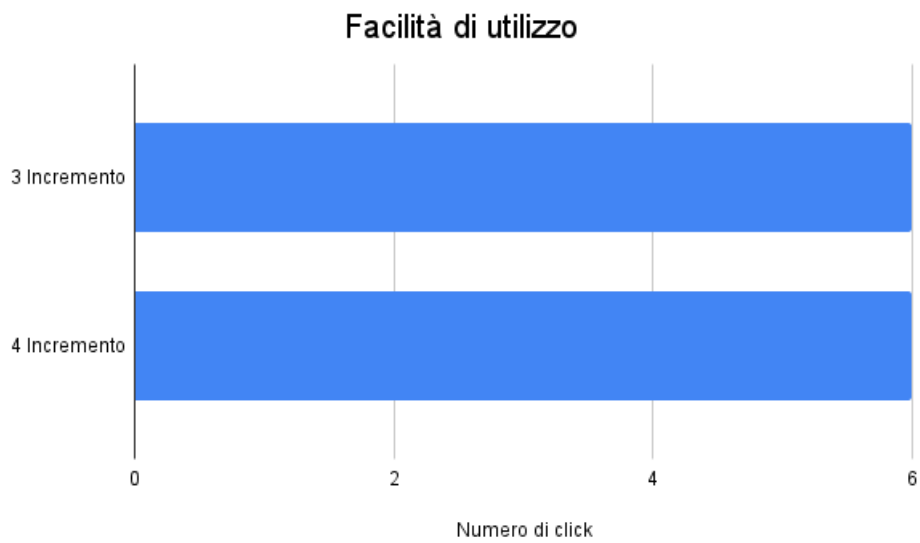
5.1.3.7 MPD07 - Facilità di utilizzo

Figura 14: Facilità di utilizzo

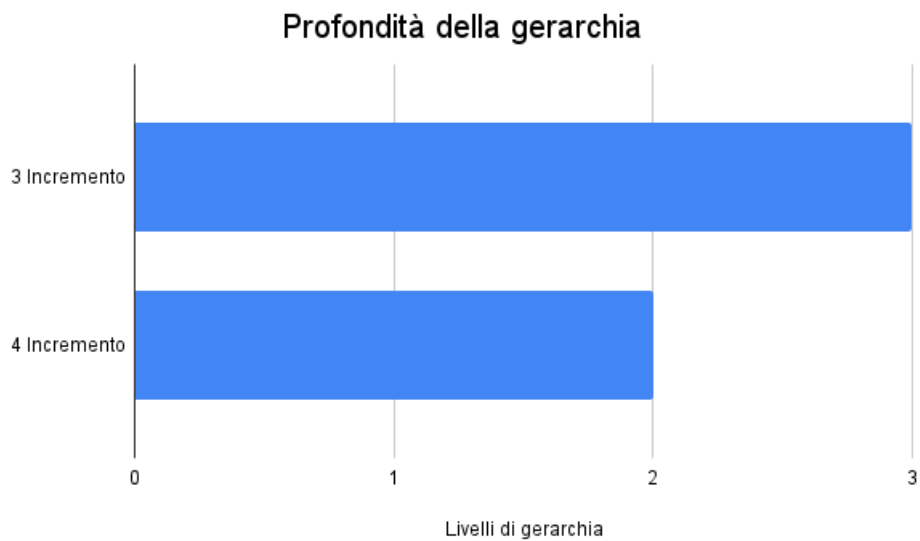
5.1.3.8 MPD08 - Profondità della gerarchia

Figura 15: Profondità della gerarchia

5.1.3.9 MPD09 - Complessità ciclomatica

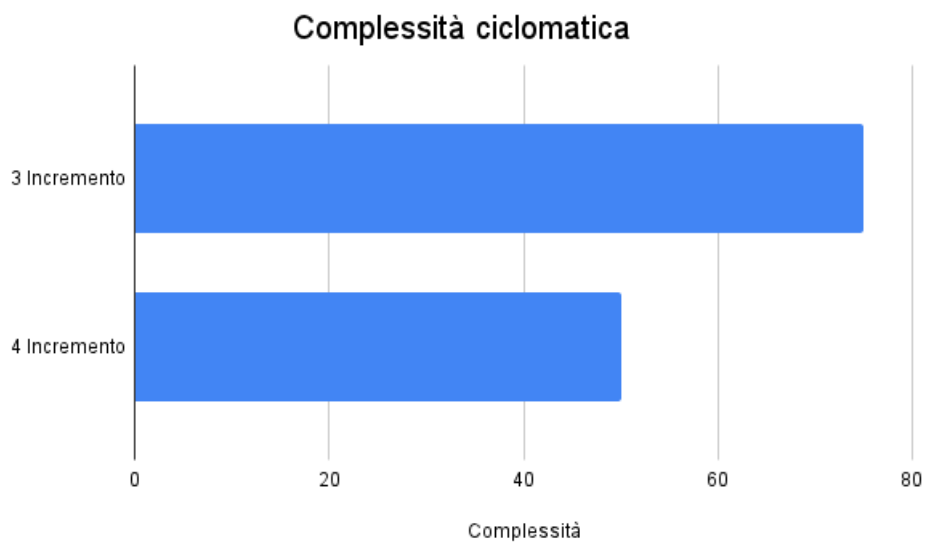


Figura 16: Complessità ciclomatica

5.1.3.10 MPD10 - Code smell

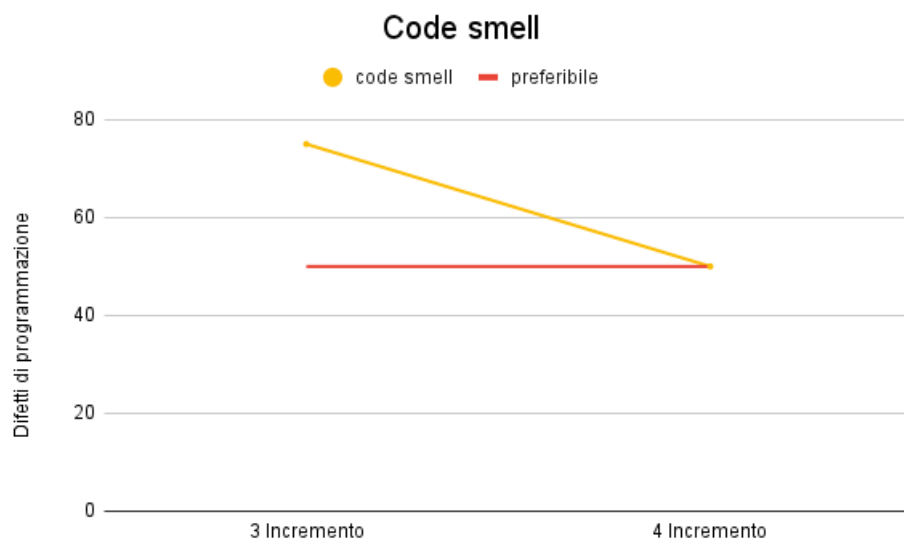


Figura 17: Code smell

5.1.3.11 MPD11 - Uso risorse

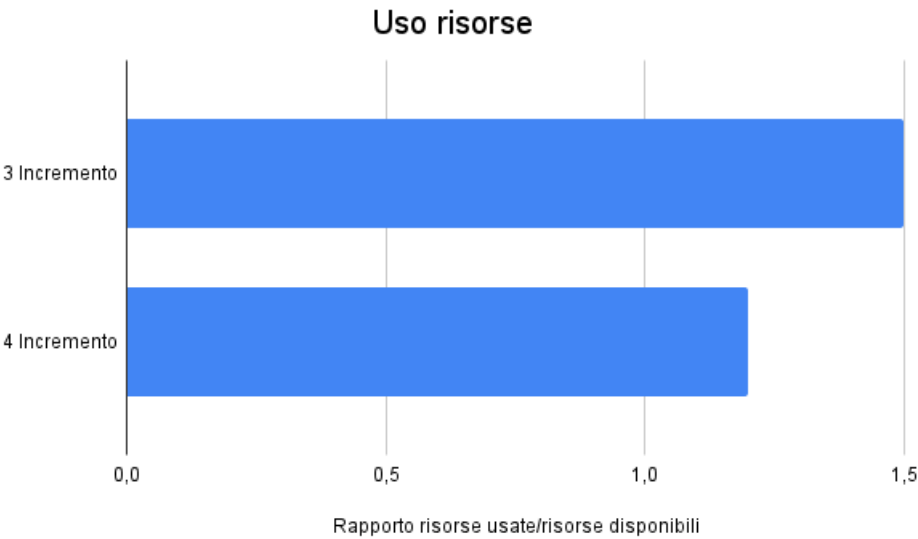


Figura 18: Uso risorse

5.1.4 Test

Di seguito si forniscono, in forma grafica, i risultati dei test indicati al capitolo 4.

5.1.4.1 Test di unità

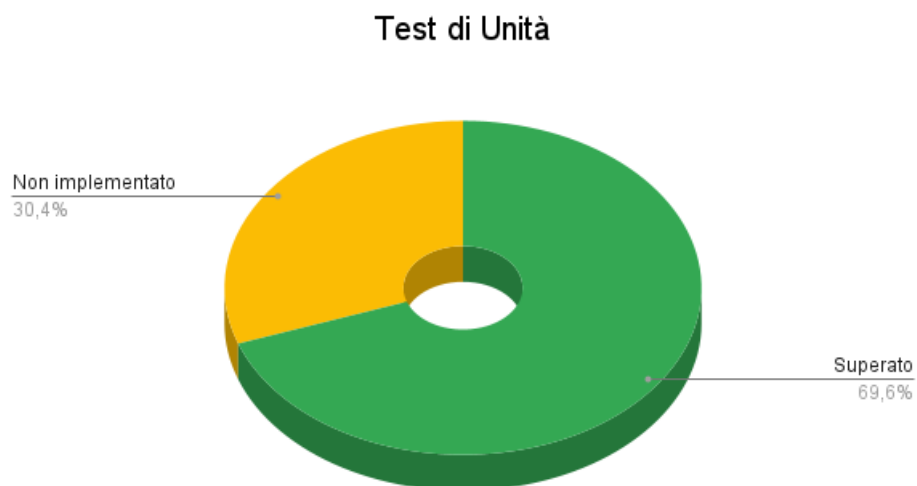


Figura 19: Risultati test di unità

5.1.4.2 Test di integrazione



Figura 20: Risultati test di integrazione

5.1.4.3 Test di sistema

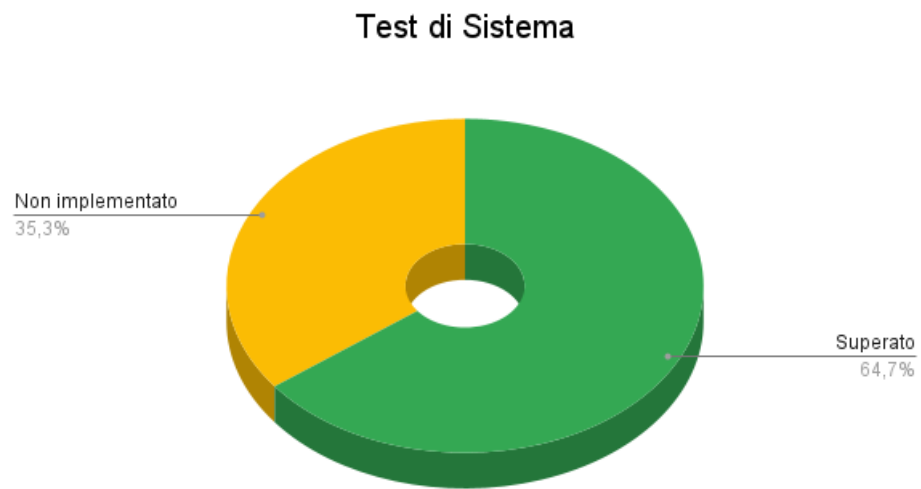


Figura 21: Risultati test di sistema

5.1.4.4 Test di accettazione

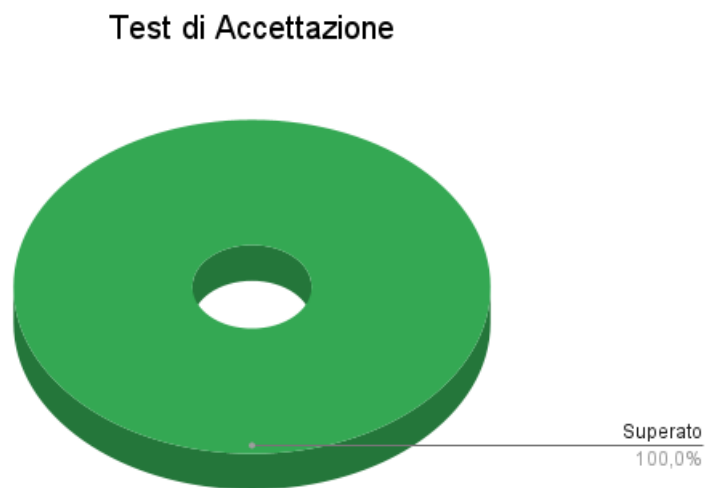


Figura 22: Risultati test di accettazione

6 Valutazioni per il miglioramento

In questa sezione viene riportata la valutazione del lavoro svolto fino ad ora. Lo scopo è quello di individuare i problemi sorti, procedere alla loro risoluzione e diminuire la probabilità che essi si verifichino nuovamente.

6.1 Organizzazione

6.1.1 Incontri di gruppo

- **Gravità:** Bassa;
- **Descrizione:** Difficoltà ad avere tutti i membri del gruppo per diversi orari lavorativi;
- **Soluzione:** Si svolgerà un *meeting online* alla settimana nell'unico momento di incontro dei membri, lunedì alle 18.

6.1.2 Organizzazione delle attività

- **Gravità:** Alta;
- **Descrizione:** Difficoltà ad intrecciare attività personali con le attività di progetto. Ciò ha portato a varie ripianificazioni delle *milestone* fissate.
- **Soluzione:** Necessario migliorare la pianificazione delle attività, tenendo conto in maniera più significativa degli impegni personali di ciascun membro.

6.1.3 Comunicazione all'interno del gruppo

- **Gravità:** Media;
- **Descrizione:** Difficoltà a comunicare con il resto del gruppo in merito ad una difficoltà. Ciò ha portato a varie attività svolte in maniera parziale o grossolana, dunque ad una nuova pianificazione dell'attività stessa, con conseguenti aumenti dei costi.
- **Soluzione:** Il responsabile sollecita settimanalmente alla comunicazione, sia durante gli incontri sia durante la settimana di lavoro.

6.2 Ruoli

6.2.1 Analista

- **Gravità:** Bassa;
- **Descrizione:** A causa dell'inesperienza del gruppo i compiti degli analisti hanno impiegato più tempo del previsto;
- **Soluzione:** Chiedere aiuto a persone più esperte per ottimizzare i tempi e non sprecare risorse.

6.2.2 Progettista

- **Gravità:** Bassa;
- **Descrizione:** A causa dell'inesperienza del gruppo i compiti dei progettisti hanno richiesto più revisioni rispetto a quanto preventivato;
- **Soluzione:** Chiedere aiuto a persone più esperte per ottimizzare i tempi e non sprecare risorse.

6.3 Strumenti di lavoro

6.3.1 Poca conoscenza generale delle tecnologie scelte

- **Gravità:** Alta;
- **Descrizione:** Lo sviluppo del prodotto richiede tecnologie che non tutti i membri conoscono;
- **Soluzione:** Formazione all'interno del gruppo e visione collettiva del codice prodotto durante la creazione del *Proof of Concept*.

6.3.2 Jira

- **Gravità:** Alta;
- **Descrizione:** Lo strumento spesso non riportava le informazioni aggiornate, in quanto chi aveva in carico una *issue* non le aggiornava;
- **Soluzione:** Il responsabile ha richiamato l'attenzione sull'importanza di aggiornare le *issue* assegnate.

6.4 Prodotto

6.4.1 Funzionalità

- **Gravità:** Alta;
- **Descrizione:** Il prodotto deve soddisfare i requisiti funzionali previsti;
- **Soluzione:** Il gruppo ha adottato l'utilizzo di *test* sulle funzionalità in modo continuo, per garantire la funzionalità.

6.4.2 Usabilità

- **Gravità:** Media;
- **Descrizione:** Il prodotto deve essere facile da usare;
- **Soluzione:** Il gruppo ha migliorato l'interfaccia utente, in modo da assecondare questa richiesta.

6.4.3 Prestazioni

- **Gravità:** Media;
- **Descrizione:** Il prodotto deve mantenere un tasso di riposta accettabile rispetto ai parametri previsti;
- **Soluzione:** Il gruppo ha analizzato e migliorato le aree del codice sorgente che causavano problemi a livello prestazionale.

6.4.4 Sicurezza

- **Gravità:** Alta;
- **Descrizione:** Il prodotto deve soddisfare dei parametri minimi di sicurezza, in modo da resistere ad attacchi da parte di soggetti malintenzionati;
- **Soluzione:** Il gruppo ha adottato vari stratagemmi per incrementare la sicurezza, resolvendo le problematiche legate ad essa ogni volta che ne sorgevano.

6.4.5 Compatibilità

- **Gravità:** Alta;
- **Descrizione:** Il prodotto deve essere compatibile con i *browser web* almeno dalla versione dichiarata;
- **Soluzione:** Il gruppo ha identificato le causa di non-compatibilità in maniera rapida e tempestiva.