



Piano di Qualifica

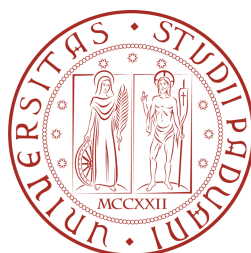
Versione 1.0.0

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



14 febbraio 2023

Contatti: dotseventeam@gmail.com

Registro delle versioni

Versione	Data	Autore	Ruolo	Motivazione
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</i>	2
2.2.2.2	<i>Actual cost of work performed</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	Obbiettivi 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	Usso Risorse	9
4	Specifica dei test	10
4.1	Test di unità	10
4.2	Test di integrazione	10
4.3	Test di sistema	10
4.4	Test di accettazione	11
4.5	Test di regressione	12
5	Resoconto delle attività di verifica	13
5.1	Processi	13
5.1.1	Fornitura	13
5.1.1.1	<i>Scheduled variance</i>	13
5.1.1.2	<i>Budget variance</i>	14
5.1.2	Sviluppo	15
5.1.3	Documentazione	15
6	Valutazioni per il miglioramento	16
6.1	Organizzazione	16
6.1.1	Incontri di gruppo	16
6.2	Ruoli	16
6.2.1	Analista	16

Elenco delle figure

1	<i>Schedule variance</i>	13
2	<i>Budget variance</i>	14
3	Indice di Gulpease	15

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 sistema	11

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. Quindi, il documento non può essere considerato definitivo nella sua attuale versione.

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>.

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>;
- Slide T09 del corso di Ingegneria del Software - Qualità di processo:
<https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T09.pdf>.

2 Qualità di processo

2.1 Introduzione

Per garantire la qualità di processo, il gruppo ha deciso di adottare lo standard **ISO/IEC/IEEE 12207:1995**. Inoltre, per garantire una corretta implementazione ed un mantenimento costante, il gruppo seguirà 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*

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*

Costo totale sostenuto per il lavoro effettivo completato fino ad ora.

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	MPC07
Gestione di qualità	Processo che consiste nel garantire gli obiettivi di qualità imposti per il prodotto	MPC08
Verifica	Processo che ha come obiettivo la valutazione della conformità o meno di un prodotto del progetto	MPC09 e MPC10

Tabella 4: Obiettivi di qualità dei processi di supporto

2.3.2 Metriche di qualità

Metrica	Nome	Valore accettabile	Valore preferibile
MPC07	Indice Gulpease	$\geq 40, \leq 100$	$\geq 60, \leq 100$
MPC08	<i>Quality Metrics Satisfied</i>	$\geq 90 \%$	100%
MPC09	<i>Code Coverage</i>	$\geq 80 \%$	100 %
MPC10	<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 che 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".

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	MQC01

Tabella 6: Obiettivi di qualità per la funzionalità

3.1.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MQC01	<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.

Tale indice si misura tramite la formula:

$$RC = \frac{Ns}{Nt} \times 100$$

Dove:

- Ns: numero di requisiti obbligatori soddisfatti;
- Nt: numero di requisiti obbligatori totali.

Per il valore ottimale, si prende Ns = 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	MQC02 MQC03
OQP03	Tolleranza agli errori	Il prodotto mantiene alte prestazioni anche in caso di malfunzionamenti o di un uso scorretto	MQC04 MQC05 MQC06

Tabella 8: Obiettivi di qualità per l'affidabilità

3.2.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MQC02	<i>Code coverage</i>	100 %	≥ 80 %
MQC03	<i>Branch coverage</i>	100 %	≥ 80 %
MQC04	<i>Failure Density</i>	0 %	15 %
MQC05	Presenza di <i>bug</i>	≤ 5	≤ 20
MQC06	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	MQC07

Tabella 10: Obiettivi di qualità per l'usabilità

3.3.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MQC07	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	MQC08
OQP06	Modificabilità	Capacità del prodotto di permettere l'implementazione di una modifica	MQC09 MQC10

Tabella 12: Obiettivi di qualità per la manutenibilità

3.4.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MQC08	Profondità della gerarchia	≤ 2	≤ 4
MQC09	Complessità ciclomatica	≤ 10	≤ 20
MQC10	<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	MQC11

Tabella 14: Obiettivi di qualità per l'efficienza

3.5.2 Metriche di qualità

Metriche	Nome	Valore Preferibile	Valore accettabile
MQC11	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 verranno stabiliti nel periodo di progettazione e codifica.

4.2 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. I test di integrazione verranno stabiliti nel periodo di progettazione e codifica.

4.3 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 ognuna delle funzionalità offerte	NI	RF01
TS-2	Si verifica che il sistema CAPTCHA riesca ad offrire la possibilità di richiedere un nuovo test CAPTCHA	NI	RF02
TS-3	Si verifica che il sistema CAPTCHA riesca a verificare la correttezza di un test CAPTCHA	NI	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	NI	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	NI	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	NI	RF07
TS-7	Si verifica che, nel caso in cui l'ID del CAPTCHA non sia valido, venga richiesto un nuovo CAPTCHA	NI	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	NI	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	NI	RF10
TS-11	Si verifica che ci sia una pagina di login protetta da CAPTCHA	NI	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	NI	RF18

Tabella 16: Test di sistema

4.4 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.

4.5 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.

5 Resoconto delle attività di verifica

5.1 Processi

5.1.1 Fornitura

5.1.1.1 *Scheduled variance*

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

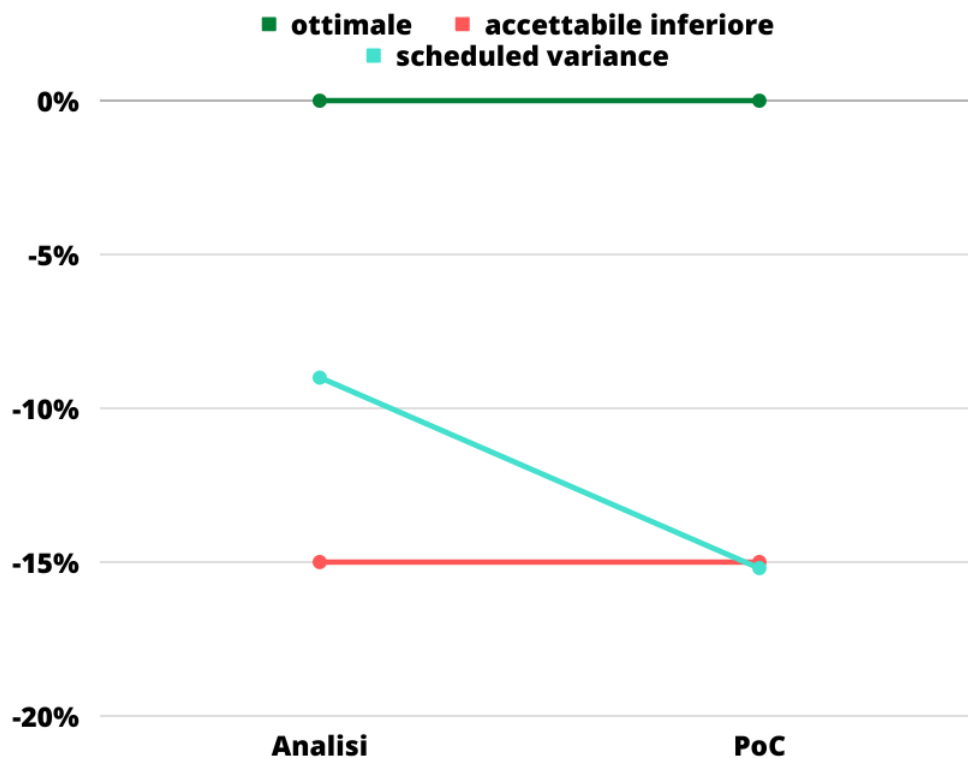


Figura 1: *Schedule variance*

Riguardo le ore preventivate, le attività del gruppo sono in ritardo, poiché si è dovuto rifare l'analisi dei requisiti. Ciò ha posticipato l'inizio della progettazione. Il grafico riguarda la *scheduled variance*. Il *team* si impegnerà al massimo per recuperare questo lieve ritardo.

5.1.1.2 Budget variance

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

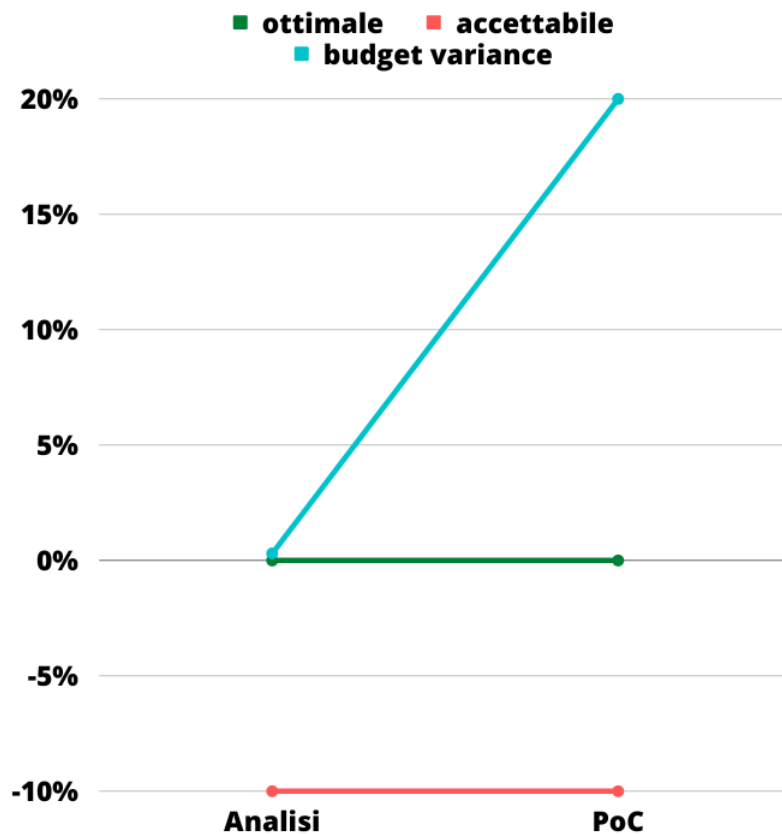


Figura 2: *Budget variance*

Il gruppo ha speso meno del previsto perché le attività sono indietro rispetto alla tabella di marcia. Questo grafico potrebbe quindi essere mal interpretato, se non viene associato a quello riguardante la *Scheduled Variance*.

5.1.2 Sviluppo

5.1.3 Documentazione

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.

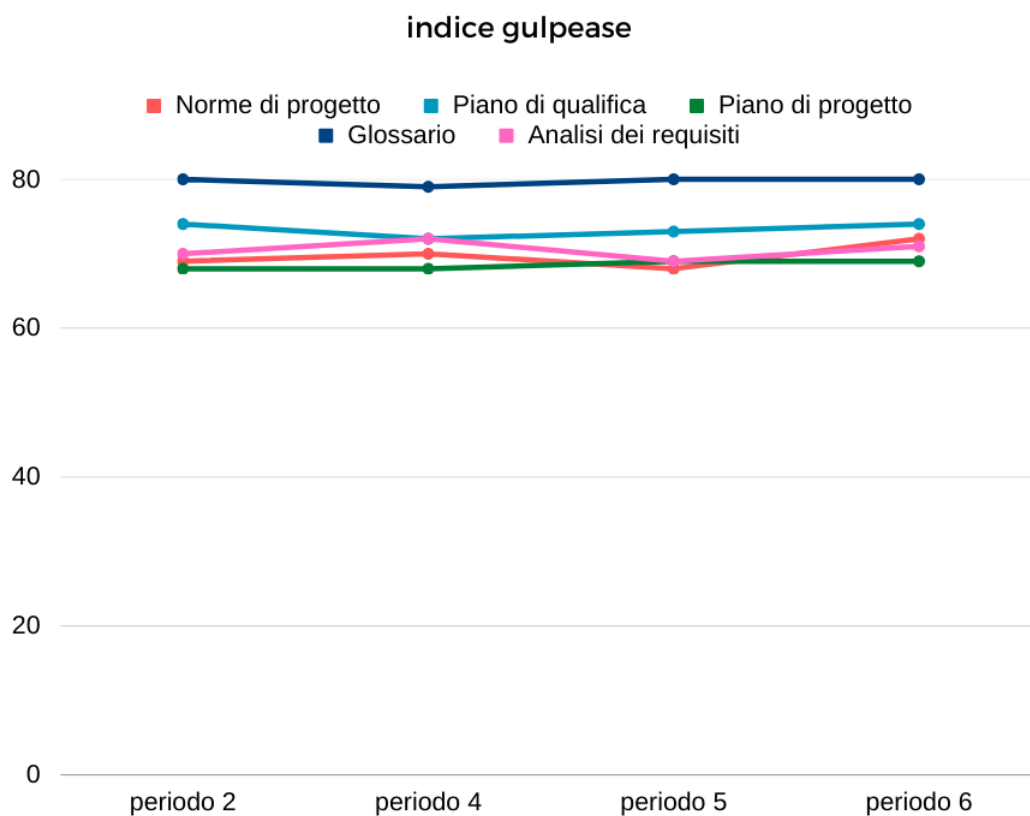


Figura 3: Indice di Gulpease

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

- **Gravita:** 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.

6.2 Ruoli

6.2.1 Analista

- **Gravita:** 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.