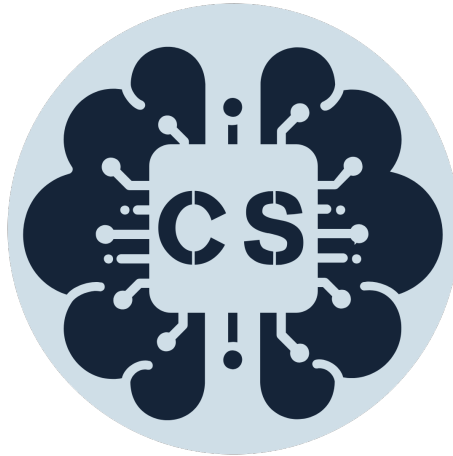


Piano di Qualifica

CyberSorcerers Team



Informazioni sul documento		
Destinatari:	Prof Tullio Vardanega	Prof Riccardo Cardin
G al pedice:	Consultare il Glossario	

Membri del team:
Sabrina Caniato
Giulia Dentone
Nicola Lazzarin
Giovanni Moretti
Andrea Rezzi
Samuele Vignotto

Registro dei Cambiamenti - Changelog_C

Versione	Data	Autore	Verificatore	Dettaglio
2.0.0	27/05/2024	Samuele Vignotto	Giovanni Moretti	Controllo finale del documento.
1.4.0	23/05/2024	Nicola Lazzarin	Andrea Rezzi	Aggiunta test di accettazione.
1.3.0	22/05/2024	Nicola Lazzarin	Andrea Rezzi	Aggiunti i test di integrazione
1.2.0	20/05/2024	Nicola Lazzarin	Andrea Rezzi	Aggiunti i test di unità.
1.1.1	06/05/2024	Giulia Dentone	Samuele Vignotto	Corretta ed estesa la sezione di Valutazione per il miglioramento.
1.1.0	04/05/2024	Samuele Vignotto	Giulia Dentone	Aggiunte conclusioni grafici.
1.0.0	09/04/2024	Samuele Vignotto	Giovanni Moretti	Verifica documento per approvazione al rilascio.
0.4.1	19/03/2024	Samuele Vignotto	Nicola Lazzarin	Aggiunta tabella test _C di sistema.
0.4.0	08/03/2024	Giulia Dentone	Samuele Vignotto	Inizio stesura della sezione testing _C .
0.3.3	22/01/2024	Sabrina Caniato	Giulia Dentone	Update della qualità di processo.
0.3.2	09/01/2024	Giovanni Moretti	Nicola Lazzarin	Update delle valutazioni per il miglioramento.
0.3.1	02/01/2024	Andrea Rezzi	Sabrina Caniato	Update delle valutazioni. per il miglioramento
0.3.0	21/12/2023	Andrea Rezzi	Sabrina Caniato	Aggiunta delle valutazioni per il miglioramento.
0.2.1	18/12/2023	Giulia Dentone	Samuele Vignotto	Update della qualità di prodotto.
0.2.0	18/12/2023	Samuele Vignotto	Giovanni Moretti	Descrizione della qualità di prodotto.

0.1.0	17/12/2023	Sabrina Caniato	Andrea Rezzi	Descrizione della qualità di processo.
0.0.1	14/12/2023	Giovanni Moretti	Nicola Lazzarin	Definizione struttura del documento e scheletro delle sezioni. Scrittura introduzione ed obiettivi delle diverse sezioni.

Indice

1	Introduzione e scopo	6
1.1	Glossario	6
1.2	Riferimenti	6
2	Qualità di processo	7
2.1	Scopo ed obiettivi	7
2.2	Processi primari	8
2.3	Processi di supporto	9
2.4	Processi organizzativi	9
3	Qualità di prodotto	9
3.1	Obiettivi	9
4	Test_G e specifiche	12
4.1	Test di Unità	12
4.2	Test di Integrazione	16
4.3	Test di Sistema	16
4.4	Test di Accettazione	18
4.5	Test di Regressione	19
5	Resoconto attività di verifica	20
5.1	Verifica documenti	20
5.1.1	Indice di Gulpease	20
5.1.2	Errori ortografici	21
5.2	Verifica processi	21
5.2.1	Stima al completamento	21
5.2.2	Valore guadagnato & valore previsto	22
5.2.3	Costo e stima al completamento	23
5.2.4	Variazione programmazione & variazione costi	23
5.2.5	Stabilità indice dei requisiti	24
5.2.6	Attualizzazione rischi	25
5.2.7	Metriche di qualità soddisfatte	25
6	Valutazioni per il miglioramento	26
6.1	Valutazione sull'organizzazione	26
6.2	Valutazione sugli strumenti utilizzati	27
6.3	Valutazione sui ruoli	28

Elenco delle figure

1	Figura 0: Modello a V	12
2	Figura 1: indice di Gulpease per periodo	20
3	Figura 2: errori ortografici per periodo	21
4	Figura 3: Revisione del valore stimato per la realizzazione del progetto	21
5	Figura 4: Valore delle attività realizzate e costo pianificato per realizzare le rimanenti	22
6	Figura 5: Costo effettivamente sostenuto e valore stimato per la realizzazione delle rimanenti attività	23
7	Figura 6: Variazione programmazione e variazione costo per periodo	23
8	Figura 7: Variazione del numero di requisiti	24
9	Figura 8: Rischi verificati per periodo	25
10	Figura 9: Metriche soddisfatte per periodo	25

Elenco delle tabelle

1	Tabella dei processi	8
2	Tabella dei processi primari	8
3	Tabella dei processi di supporto	9
4	Tabella dei processi organizzativi	9
5	Tabella degli obiettivi della qualità di prodotto	10
6	Metriche per la qualità di prodotto	11
7	Tabella dei test di sistema	18
8	Tabella dei test di sistema	19
9	Criticità sull'organizzazione	26
10	Criticità negli strumenti utilizzati	27
11	Criticità dei ruoli	28

1 Introduzione e scopo

Il Piano di Qualifica è un documento che ci prefissiamo di aggiornare periodicamente dato che definisce l'approccio e le strategie per garantire la qualità di un progetto software. Questo piano è parte integrante del processo di gestione della qualità e fornisce una linea guida dettagliata su come il controllo e l'assicurazione della qualità verranno implementati durante l'intero ciclo di vita_G del progetto.

In questo documento cercheremo di definire delle metriche di misurazione dell'efficacia e dell'efficienza del progetto, in base anche agli accorgimenti forniti dal proponente.

Il piano di qualifica conterrà:

- Definizione chiara degli obiettivi e delle metriche di qualità che il progetto propone di raggiungere.
- Specifica dei criteri che determineranno se il prodotto soddisfa gli standard di qualità stabiliti.
- Descrizione dettagliata dei processi di test_G che saranno implementati e la definizione delle strategie utilizzate per l'esecuzione di essi.
- Procedure per gestire eventuali deviazioni rispetto agli standard di qualità pianificati.

1.1 Glossario

I termini impiegati in questo testo potrebbero suscitare incertezze circa il loro significato, rendendo quindi necessaria una definizione per evitare ambiguità. Tali termini sono identificati da una lettera "G" maiuscola posta in pedice alla parola, e la loro spiegazione è fornita nel Glossario v1.0.0.

1.2 Riferimenti

Riferimenti normativi

- C7.pdf

Riferimenti informativi

- Lezione del corso di Ingegneria del software "Qualità del software":
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T7.pdf>
- Lezione del corso di Ingegneria del software "Qualità di processo":
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T8.pdf>
- Lezione del corso di Ingegneria del software "Verifica e validazione: introduzione":
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T9.pdf>
- ISO/IEC 9126:2001 SWE Product Quality
<https://www.iso.org/standard/22749.html>
- ISO/IEC 14598:1999 SW Product Evaluation
<https://www.iso.org/standard/24902.html>
- ISO/IEC 25000:2005 SQuaRE: Systems and software Quality Requirements and Evaluation
 - 25010:2011 Quality model;

- 25020:2019 Quality measurement framework;
- 25030:2007 Quality requirements;
- 25040:2011 Quality evaluation.

<https://cdn.standards.iteh.ai/samples/35683/917c6ad92a6e4c6c9326547e53f2dd7a/ISO-IEC-25000-2005.pdf>

- ISO 9000:2015

https://coopcrea.it/wp-new/wp-content/uploads/2017/10/ISO-UNI-9001_2015-Fondamenti-e-vocabolario.pdf

- ISO 9004:2018

<https://parsetrainings.com/wp-content/uploads/2018/07/ISO-9004-2018.pdf>

- ISO/IEC 33020:2019

<https://cdn.standards.iteh.ai/samples/78526/e84f5951f904440092d79e0e881c1122/ISO-IEC-33020-2019.pdf>

2 Qualità di processo

2.1 Scopo ed obiettivi

La qualità è determinata univocamente dai processi che compongono un prodotto, misurata attraverso che permettano di valutare tali processi e accertarsi che siano conformi agli obiettivi di qualità previsti. Da mettere in atto è Ciclo PDCA (Plan - Do - Check- Act)_G, che garantisce un miglioramento continuo nell'utilizzo dei processi e delle risorse tramite una prima fase di pianificazione, seguita da una verifica con le metriche previste e infine un'integrazione o correzione del prodotto in base ai risultati precedentemente ottenuti.

Obiettivo	Descrizione	Metriche
Processi primari		
Fornitura	Procedura che implica la selezione di metodologie e risorse adeguate per soddisfare le esigenze del cliente.	MC01, MC02, MC03, MC04, MC05, MC06, MC07
Sviluppo	Procedura volta a creare un prodotto software che risponda alle necessità del cliente.	MC08
Processi di supporto		
Verifica	Procedura mirata a verificare che ogni servizio realizzato soddisfi i requisiti specificati.	MC09
Gestione della qualità	Procedura volta a garantire la conformità del prodotto e dei servizi offerti agli standard predefiniti.	MC10
Processi organizzativi		
Gestione organizzativa	Procedura dedicata a delineare le modalità di coordinamento del team.	MC11

Tabella 1: Tabella dei processi

2.2 Processi primari

Codice	Nome metrica	Valore accettabile	Valore ottimale
Fornitura			
MC01	Earned Value (EV)	> 0	$\leq EAC$
MC02	Actual Cost (AC)	≥ 0	$\leq EAC$
MC03	Planned Value (PV)	≥ 0	$\leq BAC$
MC04	Cost Variance (CV)	$\geq -10\%$	$\geq 0\%$
MC05	Schedule Variance (SV)	$\geq -10\%$	$\geq 0\%$
MC06	Estimated At Completion (EAC)	$\geq BAC - 3\%;$ $\leq BAC + 3\%$	$= BAC$
MC07	Estimate To Complete (ETC)	≥ 0	$\leq EAC$
Sviluppo			
MC08	Requirements Stability Index (RSI)	$\geq 80\%$	100%

Tabella 2: Tabella dei processi primari

2.3 Processi di supporto

Codice	Nome metrica	Valore accettabile	Valore ottimale
Verifica			
MC09	Passed Tests	$\geq 80\%$	100%
Gestione della qualità			
MC10	Metrics Satisfied	$\geq 85\%$	100%

Tabella 3: Tabella dei processi di supporto

2.4 Processi organizzativi

Codice	Nome metrica	Valore accettabile	Valore ottimale
Gestione organizzativa			
MC11	Risks Found	≤ 5	0

Tabella 4: Tabella dei processi organizzativi

3 Qualità di prodotto

Per assicurare l'elevata qualità del prodotto, è stata adottata come base di riferimento la norma ISO/IEC 12207:1997. In questa sezione vengono presentati i valori ideali e quelli accettabili relativi alle metriche scelte dal team Cyber Sorceres. Per una visione dettagliata delle metriche indicate in seguito, si prega di fare riferimento al documento *Norme di progetto*.

3.1 Obiettivi

- Efficienza
- Usabilità
- Affidabilità
- Manutenibilità
- Portabilità

Obiettivo	Descrizione	Metriche
Documentazione		
Leggibilità documenti	La documentazione deve essere comprensibile agli utenti.	MD01
Correttezza linguistica	Non devono essere presenti errori grammaticali nella documentazione.	MD02
Software		
Funzionalità	La capacità del prodotto di fornire tutte le funzioni identificate nell' <i>Analisi dei requisiti</i> , perseguendo precisione e idoneità.	MS01, MS02, MS03
Usabilità	La capacità di essere comprensibile al fine di rendere gradevole l'esperienza dell'utente. Le funzionalità devono essere in linea con le aspettative e compatibili con le stesse.	MS04
Portabilità	La capacità di operare in vari contesti di esecuzione. Gli obiettivi da raggiungere includono adattabilità e sostituibilità.	MS05, MS06
Test_G	L'intero codice prodotto sarà soggetto a verifica per assicurare l'implementazione corretta dei requisiti identificati.	MS07, MS08, MS09, MS10

Tabella 5: Tabella degli obiettivi della qualità di prodotto

Codice	Denominazione metrica	Valore accettabile	Valore ottimale
MD01	Indice di Gulpease	≥ 60	≥ 80
MD02	Errori ortografici	0	0
MS01	Copertura requisiti obbligatori	100%	100%
MS02	Copertura requisiti desiderabili	$\geq 50\%$	$\geq 100\%$
MS03	Copertura requisiti opzionali	$\geq 50\%$	$\geq 100\%$
MS04	Facilità utilizzo	5 click	4 click
MS05	Versioni browser supportate	$\geq 80\%$	$\geq 100\%$
MS06	Versioni VSCode _G supportate	$\geq 80\%$	$\geq 100\%$
MS07	Solidity Statement Coverage	$\geq 80\%$	$\geq 100\%$
MS08	Solidity Branche Coverage	$\geq 80\%$	$\geq 100\%$
MS09	Solidity Function Coverage	$\geq 80\%$	$\geq 100\%$
MS10	Solidity Line Coverage	$\geq 80\%$	$\geq 100\%$

Tabella 6: Metriche per la qualità di prodotto

4 Test_G e specifiche

Nella seguente sezione esporremo le varie metodologie di test, i loro obiettivi e i criteri di successo ottenuti. Per facilitare la fase di validazione e accertamento continuo della correttezza del prodotto il gruppo ha deciso di svolgere una verifica in parallelo allo sviluppo, conformandosi al "modello a V".

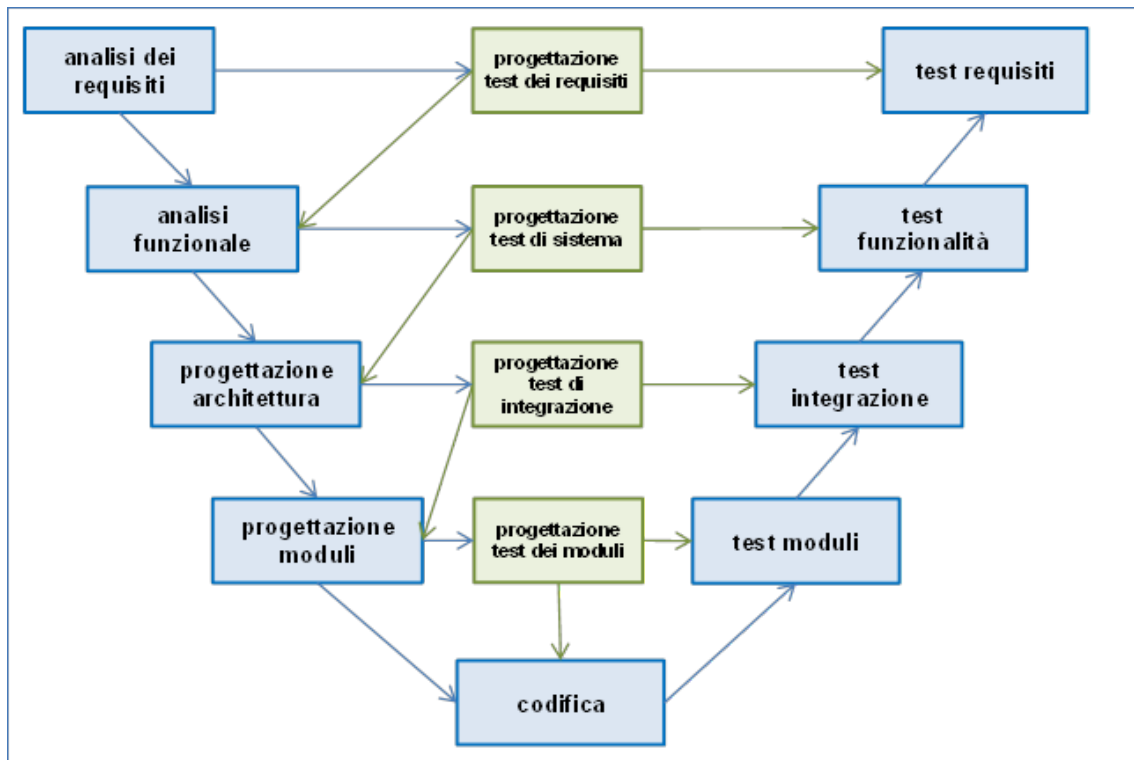


Figura 0: Modello a V

4.1 Test di Unità

Il *test di unità* sono una tipologia di testing_G del software in cui singole unità o componenti del software vengono testate in isolamento. Le unità possono essere singole funzioni, procedure, metodi o classi. L'obiettivo del test di unità è verificare che ciascuna unità funzioni correttamente secondo le specifiche e che produca i risultati attesi. Abbiamo deciso che questo tipo di testing sarà in larga parte automatizzato, per ottimizzare i costi e le tempistiche dedicate a questo processo.

ID Test _G	Descrizione	ID Requisiti	Stato
TU1	Si verifica che il componente AddProjectButton venga renderizzato correttamente		
TU2	Si verifica che il componente AddProjectButton apra una casella di inserimento testuale correttamente		
TU3	Si verifica che il pulsante di chiusura del componente AddProjectButton funzioni correttamente		
TU4	Si verifica che il pulsante di invio del componente AddProjectButton funzioni correttamente		
TU5	Si verifica che il componente BackButton venga renderizzato correttamente		
TU6	Si verifica che il componente BackButton funzioni correttamente		
TU7	Si verifica che il componente BuisnessRequest venga renderizzato correttamente	ROF2,ROF3	
TU8	Si verifica che il componente BuisnessRequest funzioni correttamente	ROF2,ROF3	
TU9	Si verifica che il componente DeleteEpic venga renderizzato correttamente		
TU10	Si verifica che il componente DeleteEpic funzioni correttamente		
TU11	Si verifica che il componente DeleteUser venga renderizzato correttamente		
TU12	Si verifica che il componente DeleteUser funzioni correttamente		
TU13	Si verifica che il componente DropdownButton venga renderizzato correttamente		
TU14	Si verifica che il componente DropdownButton funzioni correttamente	ROF14	
TU15	Si verifica che il componente DropdownMenuContainer venga renderizzato correttamente	ROF14	
TU16	Si verifica che il componente DropdownMenuContainer funzioni correttamente	ROF14	
TU17	Si verifica che il componente DropdownMenuContainer si chiuda correttamente	ROF14	

TU18	Si verifica che gli item del componente DropdownMenuContainer funzionino correttamente	ROF14
TU19	Si verifica che il componente DropdownMenu venga renderizzato correttamente	ROF14
TU20	Si verifica che il componente DropdownMenu venga funzioni correttamente	ROF14
TU21	Si verifica che il componente DropdownMenu si chiuda correttamente	ROF14
TU22	Si verifica che gli item del componente DropdownMenu funzionino correttamente	ROF14
TU23	Si verifica che il componente EpicStory venga renderizzato correttamente	ROF4, ROF17
TU24	Si verifica che nel componente EpicStory venga renderizzato il titolo della tabella correttamente	
TU25	Si verifica che nel componente EpicStory venga renderizzata la descrizione della tabella correttamente	
TU26	Si verifica che nel componente EpicStory venga renderizzato il bottone per aggiungere un'epic story correttamente	
TU27	Si verifica che il componente InviteUserButton funzioni correttamente	
TU28	Si verifica che il componente Login venga renderizzato correttamente	
TU29	Si verifica che il componente Login funzioni correttamente	
TU30	Si verifica che il componente NavigationBar venga renderizzato correttamente	
TU31	Si verifica che il componente NavigationBar funzioni correttamente	
TU32	Si verifica che il componente NotificationPage venga renderizzato correttamente	RDF6, RDF9, RDF15
TU33	Si verifica che il componente NotificationPage renderizzi il corretto numero di righe	RDF6, RDF9, RDF15

TU34	Si verifica che il componente Password venga renderizzato correttamente	ROF1
TU35	Si verifica che il toggle della visibilità nel componente Password funzioni correttamente	ROF1
TU36	Si verifica che il componente Password funzioni correttamente	ROF1
TU36	Si verifica che il componente PopupFeedback venga renderizzato correttamente	ROF5,ROF12
TU37	Si verifica che il componente PopupFeedback funzioni correttamente	ROF12
TU38	Si verifica che il componente Registrazione venga renderizzato correttamente	ROF1
TU39	Si verifica che il pulsante submit nel componente Registrazione funzioni correttamente	ROF1
TU40	Si verifica che il campo di testo email nel componente Registrazione funzioni correttamente	ROF1
TU41	Si verifica che il campo di testo password nel componente Registrazione funzioni correttamente	ROF1
TU42	Si verifica che il componente RejectProject funzioni correttamente	
TU43	Si verifica che il componente RequireAuth venga renderizzato correttamente	ROF1
TU44	Si verifica che il componente RequireAuth funzioni correttamente	ROF1
TU45	Si verifica che il componente Step1 venga renderizzato correttamente	ROF1
TU46	Si verifica che il bottone submit del componente Step1 funzioni correttamente	ROF1
TU47	Si verifica che il campo di testo email del componente Step1 funzioni correttamente	ROF1
TU48	Si verifica che il componente Step2 venga renderizzato correttamente	ROF1
TU49	Si verifica che il bottone submit del componente Step2 funzioni correttamente	ROF1
TU50	Si verifica che il campo di testo del componente Step2 funzioni correttamente	ROF1

TU51	Si verifica che il componente Table venga renderizzato correttamente	ROF4,ROF8
TU52	Si verifica che il componente Table renderizzi una riga per ogni elemento	ROF4,ROF8
TU53	Si verifica che il componente Table renderizzi un bottone correttamente	
TU54	Si verifica che il bottone del componente Table funzioni correttamente	

4.2 Test di Integrazione

I *test di integrazione* sono una fase del processo di testing_G in cui le diverse unità o del software vengono combinate e testate insieme come gruppo. L'obiettivo principale è verificare che le singole unità, testate precedentemente in modo isolato tramite i test di unità, funzionino correttamente quando integrate e collegate tra loro. Durante i test di integrazione, vengono identificati e risolti eventuali problemi di interfacciamento tra le diverse unità e vengono verificate le interazioni tra di esse. L'obiettivo finale è garantire che l'intero sistema funzioni come previsto e che tutte le interazioni tra le sue parti siano corrette.

ID Test _G	Descrizione	ID Requisiti	Stato
TU1	Verifica che ritorna il progetto con l'ID corretto _G tramite email e password	ROF1	NI
TS2	Verificare che sia possibile scrivere le richieste di business	ROF2	NI

4.3 Test di Sistema

I *test di sistema* sono una fase del processo che si concentra sull'analisi e la verifica del sistema nel suo complesso rispetto ai requisiti specificati. L'obiettivo principale è garantire che il sistema soddisfi tutte le funzionalità e i requisiti richiesti dal cliente o specificati nel documento di specifica dei requisiti. Durante i test di sistema, vengono eseguiti scenari di test realistici per simulare l'utilizzo del software in un ambiente di produzione. I risultati dei test di sistema sono utilizzati per valutare se il sistema è pronto per il rilascio o se sono necessari ulteriori miglioramenti e correzioni.

ID Test _G	Descrizione	ID Requisiti	Stato
TS1	Verificare che sia possibile accedere alla web-app _G tramite email e password	ROF1	NI
TS2	Verificare che sia possibile scrivere le richieste di business	ROF2	NI
TS3	Verificare che sia possibile inviare le richieste di business all'IA _G	ROF3	NI
TS4	Verificare che sia possibile visualizzare l'andamento del progetto dalla web-app _G	ROF4	NI
TS5	Verificare che sia possibile per l'attore _G Cliente inviare un feedback riguardo l'implementazione di una user story _G generata	ROF5	NI
TS6	Verificare che si riceva una notifica una volta che una user story _G è contrassegnata come completata	RDF6	NI
TS7	Verificare che sia possibile il tag _G del codice utilizzando il plug-in _G	ROF7	NI
TS8	Verificare che sia possibile per l'attore _G Sviluppatore il poter vedere la lista di user/epic stories _G che gli sono state assegnate	ROF8	NI
TS9	Verificare che quando viene assegnata una epic/user story _G l'attore Sviluppatore venga notificato tramite la web-app _G	RDF9	NI
TS10	Verificare che sia possibile inviare il codice all'IA _G per la verifica	ROF10	NI

ID Test _G	Descrizione	ID Requisiti	Stato
TS11	Verificare che sia possibile visualizzare le epic/user stories _G generate	ROF11	NI
TS12	Verificare che sia possibile per l'attore Project Manager _G l'invio di feedback riguardante le epic/user stories generate dall'IA _G	ROF12	NI
TS13	Verificare che sia possibile per l'attore Project Manager richiedere la divisione delle user stories valutate come troppo grandi	ROF13	NI
TS14	Verificare che sia possibile per l'attore Project Manager assegnare le epic/user stories a gli attori Sviluppatore	ROF14	NI
TS15	Verificare che venga inviata una notifica all'attore Project Manager quando vengono generate le epic/user stories	RDF15	NI
TS16	Verificare che sia possibile per l'attore Project Manager richiedere la modifica di epic/user stories all'IA	ROF16	NI
TS17	Verificare che sia possibile per l'attore Sviluppatore visualizzare l'andamento delle epic/user stories assegnate	ROF17	NI
TS18	Verificare che il plug-in _G supporti i linguaggi Typescript _G e Javascript _G	ROF20	NI
TS19	Verificare che il plug-in supporti i linguaggi Kotlin _G e Swift _G	RDF21	NI

Tabella 7: Tabella dei test di sistema

4.4 Test di Accettazione

I *test di accettazione* sono una fase finale in cui il sistema viene valutato dal cliente per determinare se soddisfa i requisiti concordati e se è pronto per il rilascio. Questi test sono orientati a verificare che il sistema sia conforme alle aspettative e alle necessità degli utenti e che sia in grado di svolgere le funzioni previste in modo efficace ed efficiente. L'obiettivo principale è confermare che il software sia pronto per essere messo in produzione e che risponda alle aspettative del cliente. I risultati dei test di accettazione sono fondamentali per prendere decisioni riguardanti il rilascio del prodotto e possono influenzare eventuali modifiche o miglioramenti futuri.

ID Test _G	Descrizione	ID Requisiti	Stato
TA1	Si verifica che l'applicazione sia in grado di gestire il recupero password, inviando opportunamente una mail di reimpostazione all'utente qualora abbia perso le proprie credenziali		S
TA2	SI verifica che il PM possa aggiungere nuovi clienti correttamente		S
TA3	Si verifica che l'applicazione sia in grado di gestire correttamente gli eventi di autenticazione dei propri tipi di utenti, inclusi login e logout	ROF13	S
TA4	Verificare che sia possibile per l'attore Project Manager assegnare le epic/user stories a gli attori Sviluppatore	ROF14	S
TA5	Verificare che venga inviata una notifica all'attore Project Manager quando vengono generate le epic/user stories	RDF15	S
TS16	Verificare che sia possibile per l'attore Project Manager richiedere la modifica di epic/user stories all'IA	ROF16	S
TA7	Verificare che sia possibile per l'attore Sviluppatore visualizzare l'andamento delle epic/user stories assegnate	ROF17	S
TA8	Verificare che il plug-in _G supporti i linguaggi Typescript _G e Javascript _G	ROF20	SI
TA9	Verificare che il plug-in supporti i linguaggi Kotlin _G e Swift _G	RDF21	S

Tabella 8: Tabella dei test di sistema

4.5 Test di Regressione

I *test di regressione* mirano a verificare che le modifiche apportate al codice sorgente o al sistema non abbiano introdotto nuovi difetti o rotto funzionalità esistenti. Questi test vengono eseguiti dopo ogni modifica al software, come aggiornamenti, correzioni di bug o nuove implementazioni. L'obiettivo è assicurarsi che le modifiche non abbiano impatti indesiderati sul comportamento del sistema, specialmente su funzionalità precedentemente testate e funzionanti correttamente. L'obiettivo del gruppo è raggiungere la massima automazione possibile dei test di regressione, al fine di ridurre i tempi di esecuzione e garantire una copertura completa dei test.

5 Resoconto attività di verifica

5.1 Verifica documenti

5.1.1 Indice di Gulpease

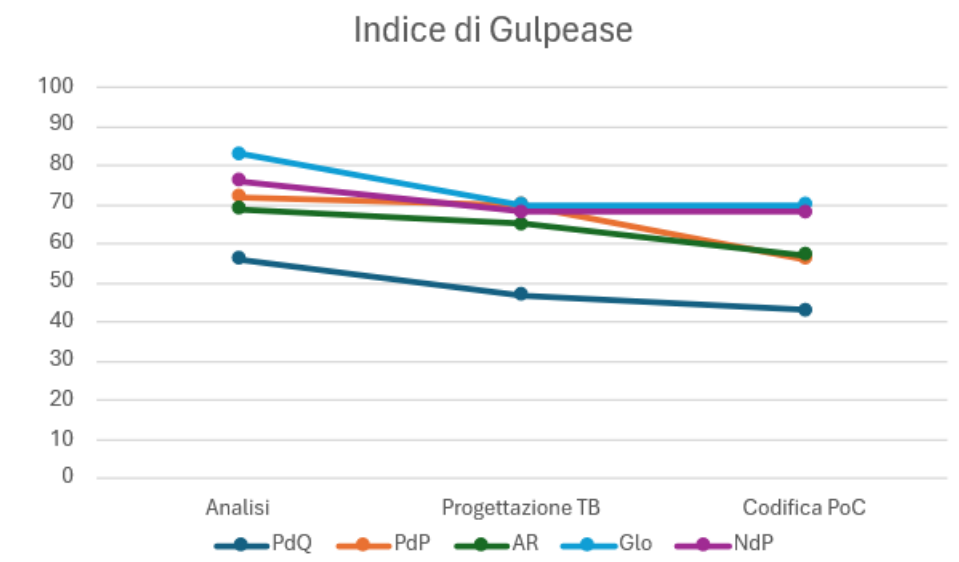


Figura 1: indice di Gulpease per periodo

L'andamento discendente dell'Indice di Gulpease nei documenti analizzati suggerisce una tendenza verso una maggiore complessità e articolazione del linguaggio nel corso del tempo. Questo declino potrebbe riflettere cambiamenti nella struttura dei testi o nell'approccio alla scrittura, sottolineando l'importanza di un'attenta valutazione delle competenze linguistiche e comunicative nel contesto attuale.

5.1.2 Errori ortografici



Figura 2: errori ortografici per periodo

Il progressivo e costante ridursi del numero di errori ortografici nei documenti analizzati dimostra un notevole miglioramento della qualità della scrittura nel tempo. Questo risultato evidenzia l'efficacia di misure correttive e di una maggiore attenzione alla precisione nell'uso della lingua.

5.2 Verifica processi

5.2.1 Stima al completamento



Figura 3: Revisione del valore stimato per la realizzazione del progetto

L'analisi del costo effettivo e della stima al completamento del progetto indica un utilizzo attuale di circa 8000€, con un saldo residuo di circa 4000€ per raggiungere il completamento totale.

Questo risultato evidenzia una gestione finanziaria attenta e prudente, mantenendo il progetto all'interno dei limiti del budget previsto. Tuttavia, è importante rimanere vigili e continuare a monitorare da vicino le spese rimanenti per garantire che il progetto sia completato in modo efficiente e senza sorprese finanziarie.

5.2.2 Valore guadagnato & valore previsto

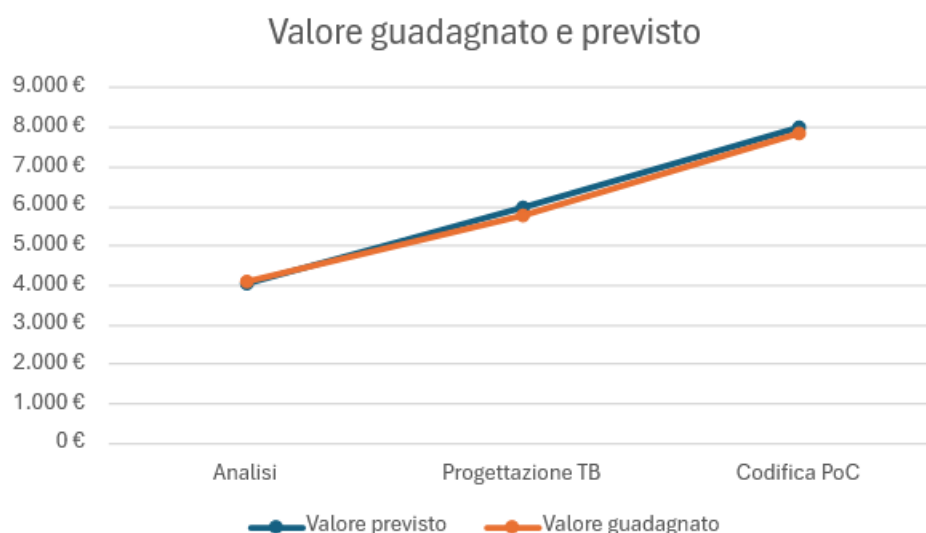


Figura 4: Valore delle attività realizzate e costo pianificato per realizzare le rimanenti

L'analisi del valore guadagnato e del valore previsto evidenzia una stretta correlazione tra le previsioni progettuali e i risultati effettivamente ottenuti, come indicato dalla quasi costante equivalenza tra i due valori nel corso del tempo. Sebbene si sia verificato un breve periodo in cui il valore guadagnato è risultato inferiore alle aspettative, è importante notare che il progetto è stato in grado di recuperare e allinearsi nuovamente con le previsioni nel periodo successivo.

5.2.3 Costo e stima al completamento



Figura 5: Costo effettivamente sostenuto e valore stimato per la realizzazione delle rimanenti attività

L'analisi del costo e della stima al completamento al periodo attuale rivela un leggero scostamento tra il budget utilizzato e quello previsto per il progetto. Mentre il budget utilizzato risulta essere leggermente più alto delle previsioni in questo punto temporale, è essenziale considerare attentamente le ragioni dietro questo scostamento e prendere le misure necessarie per mitigare eventuali ulteriori variazioni.

5.2.4 Variazione programmazione & variazione costi

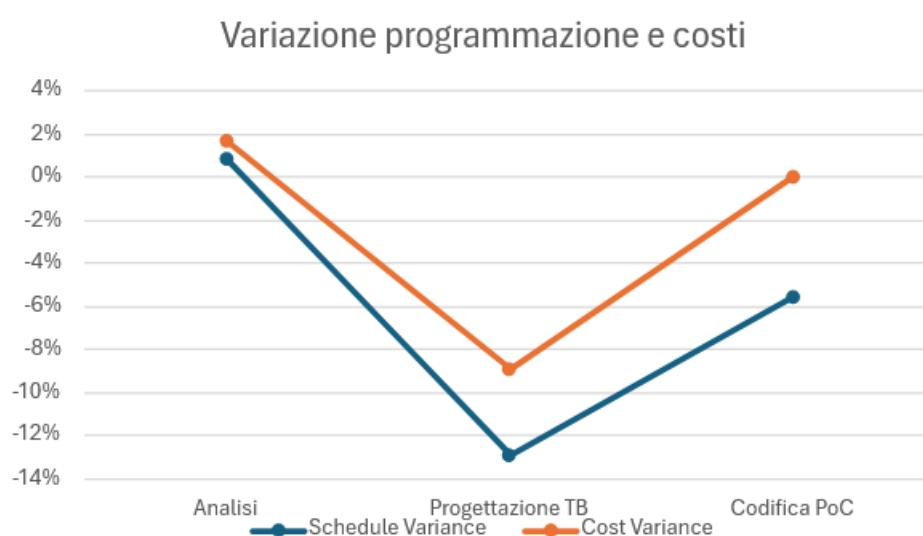


Figura 6: Variazione programmazione e variazione costo per periodo

L'analisi della variazione tra la programmazione e i costi indica che la Cost Variance è leggermente superiore alla Schedule Variance. Questo suggerisce che, sebbene il progetto sia rimasto in linea con il programma stabilito, i costi effettivi sono stati maggiori rispetto alle previsioni. Questa discrepanza sottolinea l'importanza di una gestione finanziaria più rigorosa e di un controllo più stretto sui costi nel futuro.

5.2.5 Stabilità indice dei requisiti

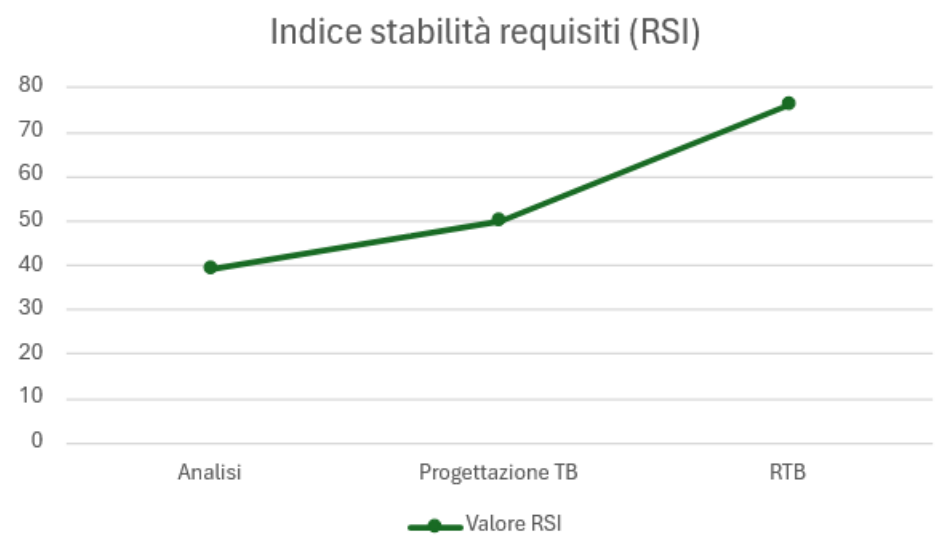


Figura 7: Variazione del numero di requisiti

L'analisi dell'Indice di Stabilità dei Requisiti (RSI) rivela un trend in aumento nel corso del tempo. Questo aumento progressivo dell'RSI indica una maggiore stabilità dei requisiti del progetto nel tempo. Tale tendenza positiva suggerisce una migliore comprensione e definizione dei requisiti, riducendo la probabilità di cambiamenti o aggiunte significative durante l'implementazione del progetto.

5.2.6 Attualizzazione rischi

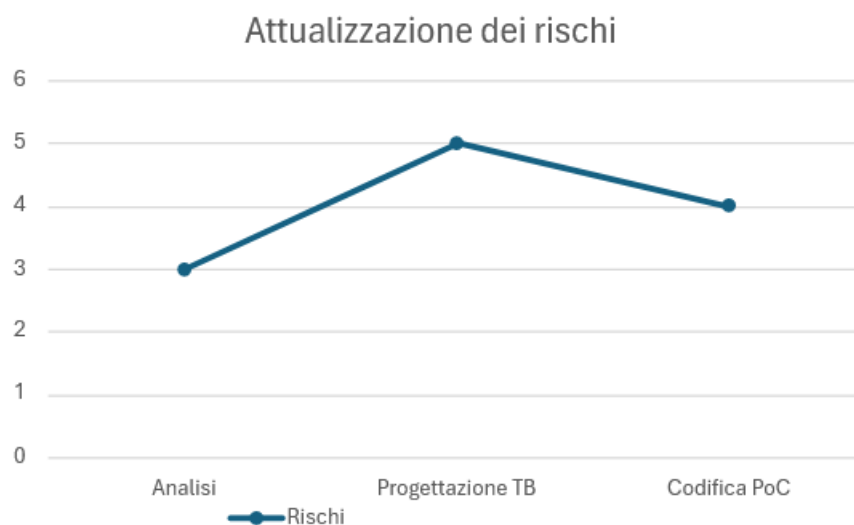


Figura 8: Rischi verificati per periodo

L'analisi dell'attualizzazione dei rischi indica un andamento variabile nel corso del tempo. Si osserva un aumento dei rischi verificatisi durante un periodo iniziale, seguito da una successiva diminuzione nel corso del tempo. Questa dinamica potrebbe riflettere l'efficacia delle azioni di mitigazione adottate per gestire i rischi identificati in fase precedente del progetto.

5.2.7 Metriche di qualità soddisfatte

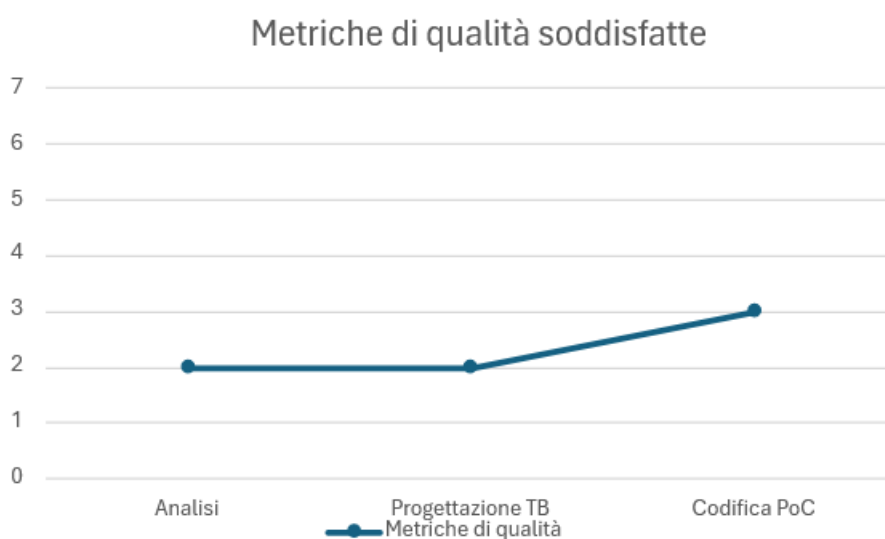


Figura 9: Metriche soddisfatte per periodo

L'analisi delle metriche di qualità soddisfatte evidenzia un trend positivo di miglioramento nel corso del tempo. Si osserva un costante aumento delle metriche soddisfatte, indicando un progressivo miglioramento della qualità nel progetto. Questo risultato può essere attribuito a una mag-

giore consapevolezza e attenzione alla qualità da parte del team, nonché all'implementazione di processi e pratiche mirate al miglioramento continuo.

6 Valutazioni per il miglioramento

In questo paragrafo, esamineremo le sfide incontrate fino alla consegna del progetto e valuteremo le soluzioni adottate dal gruppo. Nella tabella allegata, nella colonna delle soluzioni, sono elencate le strategie che abbiamo identificato per affrontare le sfide riscontrate. È importante sottolineare che il processo di miglioramento è continuo e dinamico, seguendo il principio del ciclo PDCA_G. Ciò significa che le migliorie vengono implementate in corso d'opera non appena vengono individuate criticità, consentendo al team di adattarsi e migliorare costantemente durante lo sviluppo del progetto.

6.1 Valutazione sull'organizzazione

Criticità	Descrizione	Gravità	Soluzione
Iniziale carenza di comunicazione con il cliente	Durante le prime fasi di sviluppo abbiamo avuto difficoltà ad ottenere le credenziali per utilizzare gli strumenti da loro richiesti.	Bassa	Abbiamo scelto di focalizzare il lavoro, durante l'attesa, nella redazione dei documenti e aprire un canale di comunicazione più veloce delle mail. Si è scelto Slack per la comunicazione esterna, più rapido e molto utilizzato dall'azienda. È fondamentale sviluppare da parte del team una rapida capacità di reazione.
Disparità di impegno tra i membri	Alcuni membri, avendo più impegni accademici o lavorativi, sono stati meno presenti agli incontri o per la realizzazione del progetto.	Media	Ci siamo impegnati nell'autostimare concretamente il tempo che si ha disponibile, comunicando in modo trasparente i nostri impegni settimanali (creando un calendario condiviso su Google Drive) agli altri membri, e assegnare i compiti quanto più in maniera equa e realistica.

Tabella 9: Criticità sull'organizzazione

6.2 Valutazione sugli strumenti utilizzati

Criticità	Descrizione	Gravità	Soluzione
Complessità nell'integrazione del plug-in _G	Non avendo mai sviluppato un plug-in è stata difficoltosa la fase di integrazione.	Bassa	Abbiamo scelto di focalizzarci sull'autoapprendimento di tale tecnologia e aggiungere uno Sviluppatore a discapito di altri ruoli più marginali in quella fase.
Repository _G	Difficoltà nel mantenimento dell'ordine, della linea temporale e della versioni _G dei documenti.	Media	Abbiamo focalizzato una delle fasi di verifica del Verificatore proprio sul controllo della Repository e di fare delle sedute di formazione interna per chi avesse difficoltà nell'uso delle funzionalità più utilizzate dello strumento.
Amazon AWS _G	Le librerie di Amazon AWS oltre ad essere moltissime, hanno tutte un prezzo diverso.	Alta	Abbiamo fissato un incontro di formazione da parte del proponente per scegliere in maniera mirata le librerie, in modo tale da essere conformi alle esigenze di costo e non perderci nella fase di analisi, studio dello strumento e scelta delle librerie. Ciò ha ottimizzato a livello di tempistica potenzialmente molto critica per un team inesperto.

Tabella 10: Criticità negli strumenti utilizzati

6.3 Valutazione sui ruoli

Criticità	Descrizione	Gravità	Soluzione
Verifica superficiale da parte del verificatore.	Alcuni errori sono sfuggiti durante la fase di verifica a causa di una valutazione superficiale.	Media	Abbiamo implementato checklist di verifica più dettagliate e stringenti.
Sviluppatori non allineati agli standard di codifica.	Non sempre il codice è stato conforme agli standard richiesti.	Media	Abbiamo svolto sessioni di formazione sui codici di stile e revisione del codice condiviso. Abbiamo condiviso tra membri del team materiale presente sul web fruibile e adatto allo studio individuale.
Inesperienza dell'analista.	Gli analisti, non avendo mai lavorato a un progetto di tale portata, hanno fatto fatica inizialmente ad individuare tutti i requisiti necessari dalle prime sedute.	Media	Abbiamo svolto sessioni di brainstorming interne e con i proponenti, oltre che col Prof Cardin.
Pianificazione poco realistica da parte del Responsabile.	Data l'inesperienza nell'ambito, la pianificazione e le aspettative sul carico di lavoro non sono state conformi alla realtà.	Bassa	Abbiamo definito sprint più frequenti della media, di due settimane, per mantenere sempre alta la produttività e favorire l'interazione tra i membri del team.

Tabella 11: Criticità dei ruoli

Questo approccio consente una valutazione delle sfide affrontate e delle relative soluzioni, fornendo una base più solida per il miglioramento continuo durante lo sviluppo del progetto.