

# Piano di Qualifica

CyberSorcerers Team



## **Membri del team:**

**Sabrina Caniato**

**Giulia Dentone**

**Nicola Lazzarin**

**Giovanni Moretti**

**Andrea Rezzi**

**Samuele Vignotto**

## **Informazioni sul documento**

Destinatari: Prf. Tullio Vardanega Prf. Riccardo Cardin

G al pedice: Consultare il Glossario

**Registro dei Cambiamenti - Changelog**

Versione	Data	Autore	Verificatore	Dettaglio
0.4.1	19/03/2024	Samuele Vignotto	Nicola Lazzarin	Aggiunta tabella test di sistema
0.4.0	08/03/2024	Giulia Dentone	Samuele Vignotto	Inizio stesura della sezione testing
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

## Contents

<b>1</b>	<b>Introduzione e scopo</b>	<b>4</b>
1.1	Glossario . . . . .	4
1.2	Riferimenti . . . . .	4
<b>2</b>	<b>Qualità di processo</b>	<b>5</b>
2.1	Scopo ed obiettivi . . . . .	5
2.2	Processi primari . . . . .	6
2.3	Processi di supporto . . . . .	6
2.4	Processi organizzativi . . . . .	6
<b>3</b>	<b>Qualità di prodotto</b>	<b>7</b>
3.1	Obiettivi . . . . .	7
<b>4</b>	<b>Test e specifiche</b>	<b>9</b>
4.1	Test di Unità . . . . .	9
4.2	Test di Integrazione . . . . .	9
4.3	Test di Sistema . . . . .	10
4.4	Test di Accettazione . . . . .	11
4.5	Test di Regressione . . . . .	12
4.5.1	Resoconto dei test . . . . .	12
<b>5</b>	<b>Valutazioni per il miglioramento</b>	<b>12</b>
5.1	Valutazione sull'organizzazione . . . . .	12
5.2	Valutazione sugli strumenti utilizzati . . . . .	12
5.3	Valutazione sui ruoli . . . . .	13

# 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<sub>c</sub> 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 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

- Argomento T7 - Qualità del software
- Argomento T8 - Qualità di processo
- Argomento T9 - Verifica e validazione
- - ISOG/IECG 9126:2001 SWE Product Quality;
- ISO/IEC 14598:1999 SW Product Evaluation;
- 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.
- ISO 9000:2015;
- ISO 9004:2018;
- ISO/IEC 33020:2019.

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

Table 1: Tabella dei processi

## 2.2 Processi primari

Codice	Nome metrica	Valore accettabile	Valore ottimale
<b>Fornitura</b>			
MC01	Earned Value (EV)	$> 0$	$\leq EAC$
MC02	Actual Cost (AC)	$\geq 0$	$\leq EAC$
MC03	Planned Value (PV)	$\geq 0$	$\leq BAC$
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)	$\geq 0$	$\leq EAC$
<b>Sviluppo</b>			
MC08	Requirements Stability Index (RSI)	$\geq 80\%$	100%

Table 2: Tabella dei processi primari

## 2.3 Processi di supporto

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

Table 3: Tabella dei processi di supporto

## 2.4 Processi organizzativi

Codice	Nome metrica	Valore accettabile	Valore ottimale
<b>Gestione organizzativa</b>			
MC11	Risks Found	$\leq 5$	0

Table 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
<b>Documentazione</b>		
<b>Leggibilità documenti</b>	La documentazione deve essere comprensibile agli utenti.	<b>MD01</b>
<b>Correttezza linguistica</b>	Non devono essere presenti errori grammaticali nella documentazione.	<b>MD02</b>
<b>Software</b>		
<b>Funzionalità</b>	La capacità del prodotto di fornire tutte le funzioni identificate nell' <i>Analisi dei requisiti</i> , perseguendo precisione e idoneità.	<b>MS01, MS02, MS03</b>
<b>Usabilità</b>	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.	<b>MS04</b>
<b>Portabilità</b>	La capacità di operare in vari contesti di esecuzione. Gli obiettivi da raggiungere includono adattabilità e sostituibilità.	<b>MS05, MS06</b>
<b>Test</b>	L'intero codice prodotto sarà soggetto a verifica per assicurare l'implementazione corretta dei requisiti identificati.	<b>MS07, MS08, MS09, MS10</b>

Table 5: Tabella degli obiettivi della qualità di prodotto

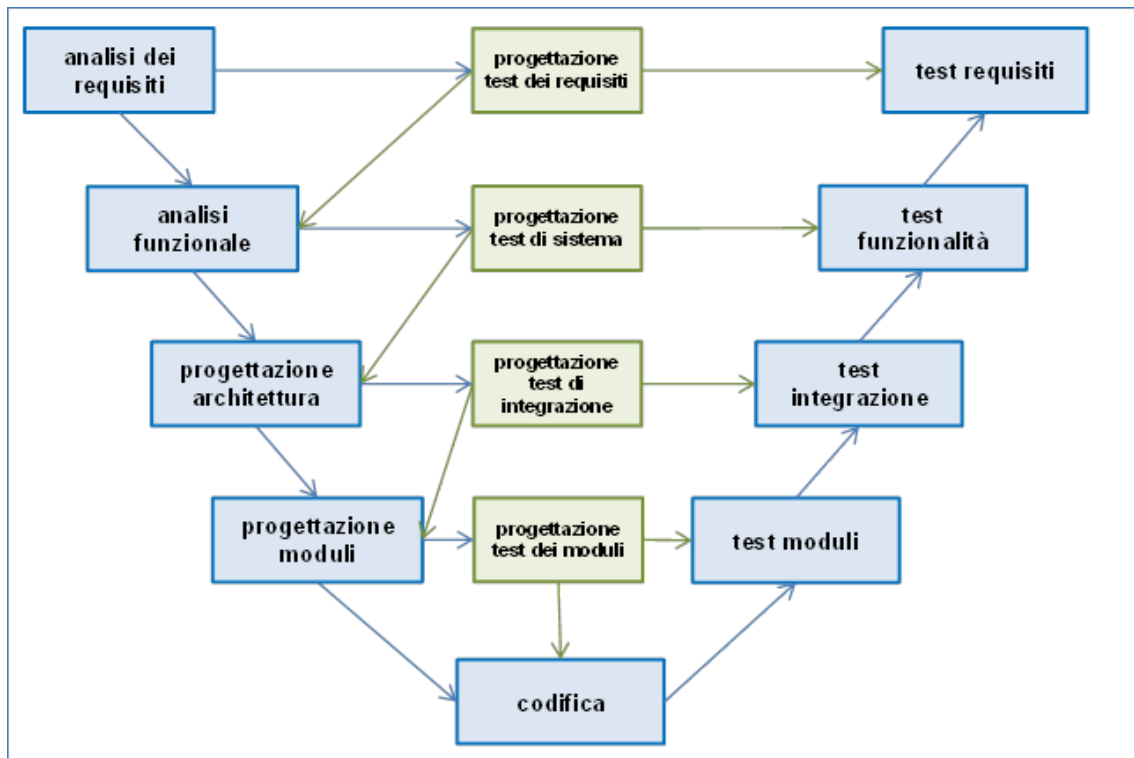
Codice	Denominazione metrica	Valore accettabile	Valore ottimale
<b>MD01</b>	Indice di Gulpease	$\geq 60$	$\geq 80$
<b>MD02</b>	Errori ortografici	0	0
<b>MS01</b>	Copertura requisiti obbligatori	100%	100%
<b>MS02</b>	Copertura requisiti desiderabili	$\geq 50\%$	$\geq 100\%$
<b>MS03</b>	Copertura requisiti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

Table 6: Metriche per la qualità di prodotto



## 4 Test 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"<sub>G</sub>.



### 4.1 Test di Unit 

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

### 4.2 Test di Integrazione

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

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

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

Table 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.

## 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.

### 4.5.1 Resoconto dei test

## 5 Valutazioni per il miglioramento

In questo paragrafo cercheremo di analizzare le difficoltà che abbiamo avuto fino alla consegna e valutarne le rispettive soluzioni e miglioramenti adottati dal gruppo.

### 5.1 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	Focalizzare il lavoro, durante l'attesa, nella redazione dei documenti e aprire un canale di comunicazione più veloce delle mail
Disparità di impegno tra i membri	Alcuni membri, avendo magari più impegni accademici o lavorativi, sono stati meno presenti agli incontri o per la realizzazione del progetto	Media	Assegnare i compiti quanto più in maniera equa e realistica

Table 8: Criticità sull'organizzazione

### 5.2 Valutazione sugli strumenti utilizzati

Criticità	Descrizione	Gravità	Soluzione
Complessità nell'integrazione del plugin	Non avendo mai sviluppato un plug in è stata difficoltosa la fase di integrazione	Bassa	Focalizzarsi sull'autoapprendimento e aggiungere uno Sviluppatore a discapito di altri ruoli più marginali in quella fase
Repository	Difficoltà nel mantenimento dell'ordine, della linea temporale e della versioni dei documenti	Media	Focalizzare una delle fasi di verifica del Verificatore proprio sul controllo della Repository e fare delle sedute di formazione interna per chi avesse difficoltà nell'uso delle funzionalità più utilizzate dello strumento
Amazon AWS	Le librerie di Amazon AWS oltre ad essere moltissime, hanno tutte un prezzo diverso	Alta	Fissare 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

Table 9: Criticità negli strumenti utilizzati

### 5.3 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	Implementazione di checklist di verifica più dettagliate
Sviluppatori non allineati agli standard di codifica	Non sempre il codice è stato conforme agli standard richiesti	Media	Sessioni di formazione sui codici di stile e revisione del codice condiviso
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	Sessioni di brainstorming interne e con i proponenti
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	Ridefinire gli sprint ed effettuarne più frequentemente ma ognuno con un minor carico di lavoro per mantenere sempre alta la produttività dei membri

Table 10: Criticità dei ruoli