



Laurea Magistrale in informatica-Università di Salerno  
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci



# Quality Plan Comun-ity

Riferimento	
Versione	1.0
Data	24/01/2023
Destinatario	Top Management
Presentato da	Testa Elio, Iannucci Michele
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
28/12/2022	0.1	Prima stesura	Testa Elio, Iannucci Michele
29/12/2022	0.2	Stesura Struttura Gestionale	Iannucci Michele
29/12/2022	0.3	Stesura Standards	Testa Elio
30/12/2022	0.4	Stesura parti restanti	Iannucci Michele, Testa Elio
30/12/2022	0.5	Revisione	Iannucci Michele
24/01/2023	1.0	Revisione finale	Testa Elio, Iannucci Michele



## Indice

---

1. Introduzione	<b>5</b>
1.1. Scopo del documento	5
1.2. Panoramica del progetto	5
2. Riferimenti	<b>5</b>
3. Struttura Gestionale	<b>6</b>
3.1. Organizzazione	6
3.2. Task	6
3.2.1. Definizione Quality Plan	6
3.2.2. Checklist di Revisione	6
3.2.3. Elaborazione Artefatto	6
3.2.4. Revisione Artefatto	7
3.2.5. Correzione Artefatto	7
3.3. Ruoli e Responsabilità	7
3.4. Comunicazione	7
3.4.1. Comunicazione sincrona	8
3.4.2. Comunicazione asincrona	9
3.4.3. Agende	9
3.4.4. Minute	9
4. Documentazione	<b>9</b>
4.1. Documentazione Team Members	9
4.2. Documentazione Project Manager	10



5. Standard, pratiche, convenzioni e metriche	<b>11</b>
5.1. Definizione della qualità	11
5.2. Standard per la documentazione	11
5.2.1. Standard per le revisioni	11
5.2.2. Standard per i documenti	11
5.3. Standard di codifica	12
5.3.1. Nomi dei file	12
5.3.2. Struttura dei file sorgente	12
5.3.3. Formattazione	12
5.3.4. Nomenclatura	13
5.3.5. Documentazione del codice	13
5.4. Standard e pratiche per il testing	13
5.5. Metriche per la valutazione del progetto	13
5.5.1. Metriche per la valutazione dei team members	13
5.5.2. Metriche per la valutazione del codice sorgente	14
6. Strumenti, tecniche e metodologie	<b>14</b>
7. Collezione, conservazione e manutenzione	<b>14</b>
8. Training	<b>15</b>



# 1. Introduzione

---

## 1.1. Scopo del documento

---

All'interno del documento mostreremo le procedure che utilizzeremo per il monitoraggio del progetto e garantire la sua qualità durante tutto lo sviluppo. Vedremo quali sono gli obiettivi di qualità del progetto, le linee guida e gli standard che seguiremo, i criteri qualitativi applicati, le procedure per l'esecuzione dei task assegnati, le metriche utilizzate per valutare e verificare la qualità del progetto, i ruoli e le responsabilità nella gestione della qualità e infine le procedure per la validazione del prodotto.

## 1.2. Panoramica del progetto

---

Il software Comun-ity è una piattaforma che si pone come obiettivo di supportare i Cittadini nei problemi quotidiani attraverso un sistema che permetta ad essi di inserire un annuncio per poter ricevere aiuto e accettare annunci di altre persone per poter dare aiuto. Il sistema rende possibile l'accreditamento di professionisti per poter accettare lavoretti e ottenere una retribuzione in un wallet virtuale, sarà inoltre possibile dare un feedback al lavoro svolto e gestire i propri annunci.

# 2. Riferimenti

---

- Standard IEEE 730/2002
- Software Project Management Plan
- Risk Management Plan
- Test Plan

## 3. Struttura Gestionale

---

### 3.1. Organizzazione

---



### 3.2. Task

---

#### 3.2.1. Definizione Quality Plan

La stesura di questo documento sarà effettuata in accordo alle indicazioni fornite dal Top Manager ed allo standard IEEE 730/2002.

#### 3.2.2. Checklist di Revisione

Per valutare la qualità dei vari artefatti saranno utilizzate le checklist fornite dai Project Manager, relative ai seguenti documenti: RAD, SDD ed ODD.

#### 3.2.3. Elaborazione Artefatto

I team members ricevono le assegnazioni dei vari task, e procedono alla realizzazione degli artefatti seguendo le indicazioni dei PM, Tutor e Top Manager.



### 3.2.4. Revisione Artefatto

---

Una volta terminata la fase di elaborazione, i team members procedono alla revisione dell'artefatto prodotto, servendosi delle checklist corrispondenti e delle indicazioni dei PM.

### 3.2.5. Correzione Artefatto

---

In caso di errori, difetti o mancanze rilevate da un team member o da un PM, si procede alla correzione dell'artefatto. In caso contrario, si procede alla sottomissione del documento.

## 3.3. Ruoli e Responsabilità

---

All'interno del progetto ci sono diverse persone che ricoprono ruoli differenti, e pertanto responsabilità diverse:

- Il Top Manager si occupa di correggere i deliverables finali che vengono consegnati in una data prestabilita
- Il Project Manager che ha il compito di gestire il team ed il progetto, monitorandone in particolare anche la qualità
- Revisore: Un team member che verifica la corretta applicazione dei criteri di qualità all'intero artefatto, prima di consegnarlo ufficialmente. La suddivisione fra i vari artefatti è la seguente:
  - **RAD:** Gabriele Santoro;
  - **SDD:** Alessandro Falcone;
  - **ODD:** Andrea Aceto;
  - **TP:** Biagio Andreucci;
  - **TCS:** Biagio Andreucci;
  - **Altri documenti di Testing:** Michele Rabesco;
  - **Implementazione:** Michele Rabesco.

## 3.4. Comunicazione

---

La struttura della comunicazione è di tipo "Peer to Peer". Ogni team member ha la possibilità di comunicare con gli altri team member e in maniera diretta anche con i 2 PM. I motivi fondamentali per cui si è scelta questo tipo di comunicazione è:



- Fornire la possibilità ad ogni team member di collaborare con un altro in caso di chiarimenti;
- Fornire una comunicazione più efficiente diminuendo l'overhead;
- Raccogliere informazioni sui team member;
- Diminuire il numero settimanali di meeting;
- Aumentare la produttività del team.

Sono stati anche individuati due tipi principali di comunicazione:

- Sincrona;
- Asincrona.

### 3.4.1. Comunicazione sincrona

Con il termine comunicazione sincrona si intende una comunicazione effettuata in tempo reale. Sono stati individuati i seguenti metodi per effettuare una comunicazione sincrona:

- Meeting;
- Teams;
- Telefonate.

**Meeting:** è stato previsto almeno un meeting a settimana concordato ad inizio progetto con il Top Manager e comunicato a tutti i membri del team il cui scopo è quello di:

- chiarire eventuali dubbi da parte di uno o più team member;
- segnalare problemi;
- assegnare Task ai team member;
- dare consigli al team per migliorare eventuali problemi dovuti alla scarsa comunicazione.

Il giorno prima di ogni meeting verrà inviata tramite la piattaforma “Trello” l’agenda contenente l’aula, l’orario e tutti i punti che saranno discussi. È inoltre possibile convocare altri meeting in maniera straordinaria per comunicazioni particolari.

**Teams:** Sono stati condivisi i contatti Teams di ogni team member e dei due PM per effettuare videochiamate nel caso non sia possibile effettuare meeting di persona.





**Telefonate:** Sono stati condivisi tutti i numeri di telefono di ogni team member e dei due PM per effettuare comunicazioni urgenti.

### 3.4.2. Comunicazione asincrona

Con il termine comunicazione asincrona si intendono delle comunicazioni che non richiedono il collegamento diretto con l'interlocutore. Il principale metodo adottato per questo tipo di comunicazione è il software cross platform "Slack" che permette un sistema di chat tramite inviti in canali. È stato creato un canale di comunicazione generale dove tutti i team member e i PM hanno accesso e un canale per ogni documento e fase del progetto per discutere di eventuali problematiche senza ricorrere ad un meeting. Se la conversazione effettuata non risulta essere esaustiva nella comprensione si provvederà ad effettuare una discussione durante il meeting settimanale oppure in un meeting straordinario se necessario.

### 3.4.3. Agende

L'Agenda è un documento contenente informazioni su task e discussioni da effettuare durante il meeting. Questo documento verrà caricato 1 giorno prima del meeting per informare i Team Members sulle discussioni che avverranno durante il meeting. Tutte le Agende seguiranno questo identificatore: Agenda\_gg\_mm\_aaaa.

### 3.4.4. Minute

La Minuta è un documento redatto per descrivere in maniera sintetica tutte le decisioni prese durante il meeting e tutte le task assegnate e concluse. Tale documento è redatto da un Team Member a rotazione e avrà scadenza del giorno successivo a quello del meeting. Tutte le Minute saranno caricate su Trello e seguiranno questo identificatore: Minuta\_gg\_mm\_aaaa.

## 4. Documentazione

### 4.1. Documentazione Team Members

Documento	Descrizione
RAD	Documento principale della fase di analisi e



	raccolta dei requisiti, che include scenari, casi d'uso e mock-up.
SDD	In questo documento comincia la fase di progettazione, in cui il sistema viene partizionato in sottosistemi.
ODD	Documento che analizza le componenti a basso livello del sistema.
TCS	Documento di specifica dei test cases.
TP	Documento di pianificazione dei test
Matrice di tracciabilità	Documento che permette di tracciare uno specifico requisito funzionale fra i vari documenti
TIR	Documento che contiene informazioni su test che sono andati in successo
TSR	Documento che contiene informazioni sui test eseguiti
TIP	Documento che contiene la pianificazione del test d'integrazione
TER	Documento che contiene informazioni sul test di sistema
Manuale d'uso	Documento che contiene informazioni su come utilizzare il sistema creato
Manuale d'installazione	Documento che contiene informazioni sull'installazione del sistema

## 4.2. Documentazione Project Manager

Documento	Descrizione
Business case	Documento di analisi finanziaria dell'intero progetto.
Post Mortem Review	Questionario teso a valutare il successo del progetto, evidenziando le lessons learned.
Project Charter	Documento che formalizza l'avvio del progetto.
Quality Plan	Documento di specifica dei criteri di qualità.
Stakeholder register	Registro di tutte le persone coinvolte in maniera diretta o indiretta nel progetto.



SMP	Documento che dettaglia la pianificazione dei task.
SPMP	Documento che illustra l'intera gestione del progetto.
Scope Statement	Documento che spiega lo scope del progetto e le relative funzionalità supportate.
Team Contract	Documento che dettaglia il contratto firmato dal team.
Test Plan	Documento che pianifica l'intera fase di testing.
WBS	Documento che illustra la scomposizione degli artefatti in singole task.
WBS Dictionary	Documento che contiene informazioni specifiche sui vari work package
SOW	Documento che illustra la proposta di progetto.
RMP	Documento che illustra il piano di gestione dei rischi.
Status Reports	Documento periodico di aggiornamento sullo stato di avanzamento del progetto.

## 5. Standard, pratiche, convenzioni e metriche

### 5.1. Definizione della qualità

Per quanto riguarda la definizione di qualità del software si fa riferimento allo standard ISO/IEC 9126. Questo si divide in quattro parti:

- Modello della qualità del software: classificato da sei caratteristiche generali e varie sotto-caratteristiche misurabili attraverso delle metriche.
- Metriche esterne per la qualità: misurano i comportamenti del software sulla base dei test, dell'operatività e dall'osservazione durante la sua esecuzione, in funzione degli obiettivi stabiliti.
- Metriche interne per la qualità: si applica al codice sorgente, durante la fase di codifica.
- Metriche per la qualità in uso: rappresenta il punto di vista dell'utente sul software e il livello di qualità in uso è raggiunto quando il livello di qualità esterna e interna sono raggiunti.



## 5.2. Standard per la documentazione

---

### 5.2.1. Standard per le revisioni

---

Ogni documento conterrà la Revision History aggiornata con ogni modifica che verrà apportata con l'obiettivo di tenere traccia della versione, della modifica e dell'autore di tale modifica. Ogni modifica sarà revisionata e riadattata al contesto del documento. In fase di revisione nel caso in cui si individuano eventuali difetti si procederà alla modifica, altrimenti si procederà alla consegna.

### 5.2.2. Standard per i documenti

---

Sarà cura del Project Manager definire la base del documento, contenente la struttura e gli stili da applicare. I Team Members saranno informati sullo stile da utilizzare per permettere loro di rispettarlo e di effettuare eventuali revisioni.

## 5.3. Standard di codifica

---

### 5.3.1. Nomi dei file

---

Per quanto riguarda la nomenclatura della documentazione avrà la seguente identificazione C17\_<Sigla del Documento>\_Versione per i documenti dei Team Members, mentre per i documenti dei Project Manager avranno il seguente identificativo: <Sigla del Documento>\_Comun-ity\_Versione.

Invece, per quanto riguarda la nomenclatura dei file del codice sorgente fare riferimento al documento C17\_ODD.

### 5.3.2. Struttura dei file sorgente

---

La struttura del file sorgente è dettata dall'utilizzo del linguaggio di programmazione Java quindi dalle convenzioni della Sun e dall'utilizzo di Maven.

### 5.3.3. Formattazione

---

Per la formattazione del codice sorgente si seguiranno le convenzioni definite dalla Sun.

Mentre, per quanto riguarda i componenti HTML:

- Ogni tag di apertura deve essere necessariamente seguito dall'apposito tag di chiusura, eccezione fatta per i tag self-closing (es. <hr>,<br></br>,<img>,...);



- Il blocco di codice dev'essere opportunamente indentato;
- L'indentazione del codice deve avvenire tramite tabulazioni (tasto TAB) e non tramite i classici spazi bianchi (tasto BARRA DI SPAZIATURA);
- Il codice dev'essere tutto scritto in lowercase, es. `<hr>` e non `<HR>`;
- I tag `<script>` devono essere posizionati alla fine del file (in genere questi vanno posizionati prima del tag di chiusura);

CSS:

- Il CSS non dev'essere definito all'interno dei tag HTML, bensì dev'essere definito all'interno dei fogli di stile. (Eccetto per casi particolari in cui è richiesto un cambiamento di una singola componente)
- Le regole CSS devono essere scritte su più righe
- Utilizzare le shorthand ove possibile.

#### 5.3.4. Nomenclatura

---

È buona norma utilizzare nomi che siano:

- Descrittivi o Non troppo lunghi o Non abbreviati o Pronunciabili;
- I nomi delle variabili devono essere scritti secondo il Camel Case: devono iniziare con lettera minuscola e le parole seguenti con la lettera maiuscola (per identificare appunto l'inizio di una nuova parola), es. `myArray`. Le variabili dovranno essere definite all'inizio del blocco di codice;
- Le variabili costanti seguiranno invece la notazione Macro Case: devono utilizzare soltanto lettere maiuscole, separate dal trattino basso: es. `ARRAY_SIZE`;
- Anche i metodi devono essere scritti secondo il Camel Case. Questi, in genere, sono formati da verbo + nome oggetto: il verbo identifica l'azione da compiere sull'oggetto, es. `getUsername`;
- Il codice deve essere provvisto di commenti per facilitare la lettura e la comprensione. Questi dovranno descrivere la funzionalità oggetto;
- I nomi delle classi e delle pagine devono invece essere scritti secondo il Capital Camel Case: devono iniziare con lettera maiuscola, così come le parole che seguiranno, es. `ServletLogin.java`;
- I nomi dei package devono essere scritti in Lower Case: devono utilizzare soltanto lettere minuscole: es. `account`.



### 5.3.5. Documentazione del codice

---

Il codice sarà costantemente commentato secondo lo standard richiesto JavaDoc e, a termine del progetto, sarà generata una documentazione del codice attraverso l'utilizzo di questo.

## 5.4. Standard e pratiche per il testing

---

Fare riferimento al documento di Test Plan.

## 5.5. Metriche per la valutazione del progetto

---

### 5.5.1. Metriche per la valutazione dei team members

---

I Team Members verranno valutati con cadenza settimanale secondo i seguenti criteri di valutazione:

- Proattività: verrà valutata la capacità di risoluzione dei Team Members alle problematiche;
- Qualità del lavoro: verrà valutata la qualità degli artefatti realizzati dai Team Members;
- Produttività: verrà valutata la produttività dei Team Members in base agli artefatti realizzati;
- Teamwork: verrà valutata la capacità dei Team Members di lavorare in gruppo per raggiungere obiettivi comuni di progetto, rispettando le scadenze;
- Comportamento ai meeting: verrà valutata la partecipazione e il comportamento ai meeting dei Team Members.

La votazione sarà espressa in una scala da 0 a 10. La valutazione finale verrà ottenuta attraverso la media delle votazioni ricevute durante l'arco di tutto il progetto.

### 5.5.2. Metriche per la valutazione del codice sorgente

---

- Il numero di warning sollevati da Checkstyle deve essere basso;
- Per ciò che riguarda il testing, devono essere testate esattamente 5 funzionalità.

## 6. Strumenti, tecniche e metodologie

---

Strumenti utilizzati:

- Template appositi preparati dal Project Manager, aventi struttura e stili da utilizzare per la stesura dei documenti;
- Linee guida per la realizzazione degli artefatti e del codice in fase di implementazione;



- Checklist per il controllo della qualità e della correttezza di ogni documento;
- Checkstyle per il controllo della correttezza del codice sorgente;
- GitHub Action per il controllo delle modifiche e dell'intero sistema in fase di sviluppo per ogni commit e pull request effettuate sui vari branch.

Il processo di controllo della qualità avviene ogni volta che un team member consegna il proprio task su Trello. Successivamente, i PM provvederanno a revisionare il task e lasciare il proprio feedback durante il meeting settimanale. Inoltre, ogni settimana i team member riceveranno una valutazione sulla qualità degli artefatti prodotti e sulla produttività. Le valutazioni verranno consegnate ai team member ogni settimana, in modo da permettergli di migliorarsi nelle settimane successive di lavoro.

## 7. Collezione, conservazione e manutenzione

---

Sia la documentazione che il codice sorgente saranno gestiti tramite Git e hostato in remoto su GitHub in una repository pubblica, disponibile al seguente indirizzo: <https://github.com/eliotest98/Comun-ity>

## 8. Training

---

È stato effettuato Training su GitHub e su MongoDB in modo da fornire ai Team Members tutte le informazioni necessarie per prepararsi all'implementazione.