

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| QP  Quality Plan  CoralloSmart   |  |  | | --- | --- | | Riferimento |  | | Versione | 0.1 | | Data | 15/11/2022 | | Destinatario | ARPA | | Presentato da | Gerardo Iuliano e Antonio Trovato | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 15/11/2022 | 0.1 | Prima stesura | Gerardo Iuliano, Antonio Trovato |
| 20/11/2022 | 0.2 | Stesura completa | Gerardo Iuliano, Antonio Trovato |

Sommario

[RevisionHistory 2](#_Toc122693573)

[Quality Plan CoralloSmart 5](#_Toc122693574)

[1. Introduzione 5](#_Toc122693575)

[1.1. Scopo del documento 5](#_Toc122693576)

[1.2. Evoluzione del documento 5](#_Toc122693577)

[2. Riferimenti 5](#_Toc122693578)

[3. Struttura gestionale 5](#_Toc122693579)

[3.1. Organizzazione 5](#_Toc122693580)

[3.2. Task 6](#_Toc122693581)

[3.2.1. Quality Plan 6](#_Toc122693582)

[3.2.2. Checklist di revisione 6](#_Toc122693583)

[3.2.3. Elaborazione artefatto 6](#_Toc122693584)

[3.2.4. Revisione artefatto 6](#_Toc122693585)

[3.2.5. Correzione artefatto 6](#_Toc122693586)

[3.3. Ruoli e responsabilità 6](#_Toc122693587)

[4. Documentazione 7](#_Toc122693588)

[5. Standard, metriche e convenzioni 7](#_Toc122693589)

[5.1. Definizione della qualità 7](#_Toc122693590)

[5.2. Standard per la documentazione 11](#_Toc122693591)

[5.2.1. Standard di revisione 11](#_Toc122693592)

[5.2.2. Standard per i documenti 11](#_Toc122693593)

[5.2.3. Standard per lo scambio di documenti 12](#_Toc122693594)

[5.3. Standard per gli artefatti 12](#_Toc122693595)

[5.4. Standard di codifica 12](#_Toc122693596)

[5.4.1. Nomi dei file 12](#_Toc122693597)

[5.4.2. Struttura dei file sorgente 12](#_Toc122693598)

[5.4.3. Formattazione 12](#_Toc122693599)

[5.4.4. Dichiarazioni 12](#_Toc122693600)

[5.4.5. Nomenclatura 13](#_Toc122693601)

[5.4.6. Documentazione del codice 13](#_Toc122693602)

[5.4.7. Base di dati 13](#_Toc122693603)

[5.5. Standard e pratiche per il testing 14](#_Toc122693604)

[5.6. Metriche per la valutazione 14](#_Toc122693605)

[5.6.1. Metriche di valutazione dei Team Member 14](#_Toc122693606)

[5.6.2. Metriche di valutazione del codice 15](#_Toc122693607)

[6. Revisione del software 16](#_Toc122693608)

[7. Test 16](#_Toc122693609)

[8. Rapporti sui problemi e azioni correttive 16](#_Toc122693610)

[9. Strumenti, tecniche e metodologie 16](#_Toc122693611)

[10. Controllo dei dati multimediali 17](#_Toc122693612)

[11. Controllo della fornitura 17](#_Toc122693613)

[12. Collezione e conservazione dei dati 17](#_Toc122693614)

[13. Training 17](#_Toc122693615)

[14. Gestione dei rischi 17](#_Toc122693616)

[15. Glossario ed Acronimi 17](#_Toc122693617)

[16. Procedure di aggiornamento del piano 18](#_Toc122693618)

Quality Plan  
CoralloSmart

1. Introduzione
   1. Scopo del documento

Nel documento verrà descritta l’attività di monitoraggio e la gestione della qualità, entrambe eseguite per tutta la durata del progetto. Saranno definiti gli obiettivi di qualità, i criteri, gli standard e le linee guida da utilizzare, le metodologie scelte, metriche e tecniche per definire la qualità degli artefatti da produrre.

Faremo riferimento allo standard IEEE 730/2002 nei limiti dovuti alla natura didattica del progetto.

* 1. Evoluzione del documento

Tale documento subirà aggiornamenti quando necessario.

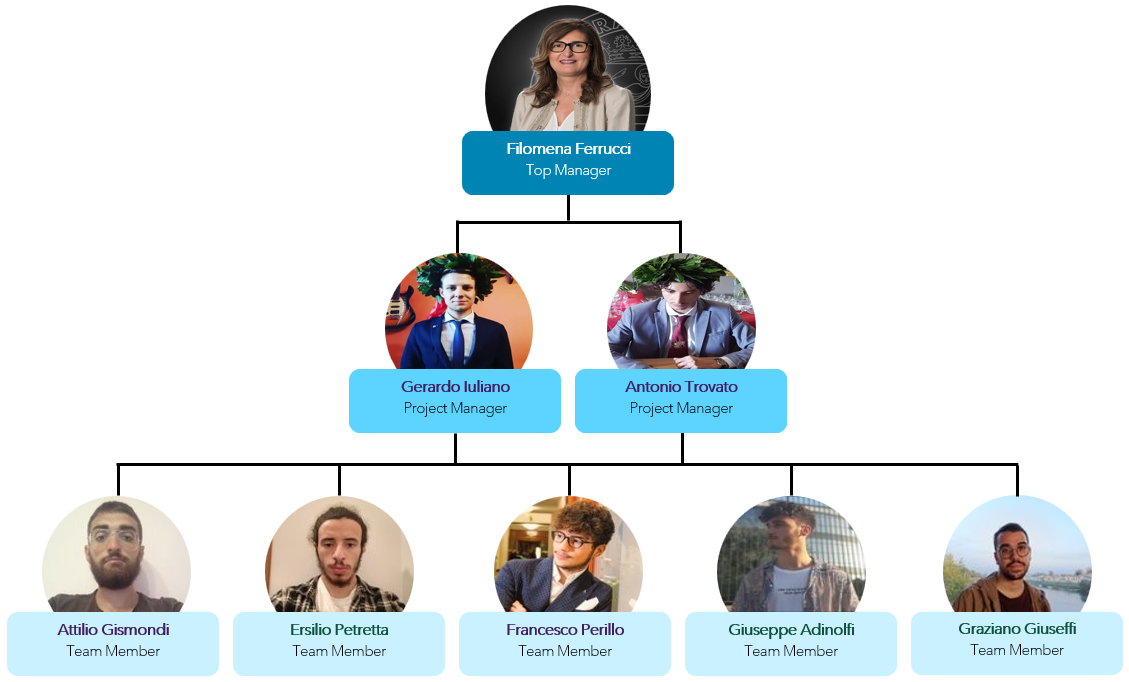
1. Riferimenti

Faremo riferimento ad ulteriori documenti quali:

* Software Project Management Plan
* Risk Management Plan
* Team Contract

1. Struttura gestionale
   1. Organizzazione

Il progetto CoralloSmart ha un’organizzazione di tipo gerarchica. In alto è presente il Top Manager che supervisionerà l’intero progetto durante tutte le sue fasi comunicando con i Project Manager. I PM invece coordineranno il team.



* 1. Task
     1. Quality Plan

Il Quality Plan tiene conto degli standard di qualità definiti dal Top Manager e fa riferimento allo standard IEEE.

* + 1. Checklist di revisione

Le checklist adottate sono state fornite dal Top Manager e riguardano RAD, SDD e ODD.

Sono state fornite ai Team Member tramite la piattaforma Teams.

* + 1. Elaborazione artefatto

Dopo aver ricevuto i task, i Team Member potranno produrre degli artefatti che siano conformi alle checklist fornite. Le checklist forniranno una linea da seguire e la possibilità di autovalutare la qualità di quanto prodotto.

* + 1. Revisione artefatto

Gli artefatti prodotti saranno ispezionati in un primo momento dai Team Member incaricati alla revisione. Successivamente i Project Manager avranno il compito di ispezionare, scovare e annotare i possibili difetti.

* + 1. Correzione artefatto

Al termine della revisione ci sono diversi scenari:

Se non sono stati riscontrati difetti nell’artefatto prodotto, allora esso sarà definito come definitivo e in caso di modifiche necessarie, bisognerà fare richiesta ai PM.

Se sono stati riscontrati difetti nell’artefatto prodotto, allora risulterà necessario effettuare un lavoro di re-work affidato al responsabile del task. Infine, sarà eseguita un nuovo confronto con le checklist fornite.

* 1. Ruoli e responsabilità
* **Top Manager**: ha il ruolo di accettare i deliverable prodotti.
* **Project Manager**: ha la responsabilità (accountability) di verificare la corretta applicazione di quanto definito nelle prossime sezioni del documento.
* **Team Member**: ha il compito di portare a termine il task che gli è stato assegnato. Nel caso di Team Member revisore, ha anche il compito di effettuare una prima revisione del documento e/o artefatto.

I Team Member addetti alle revisioni sono: Graziano Giuseffi e Francesco Perillo.

1. Documentazione

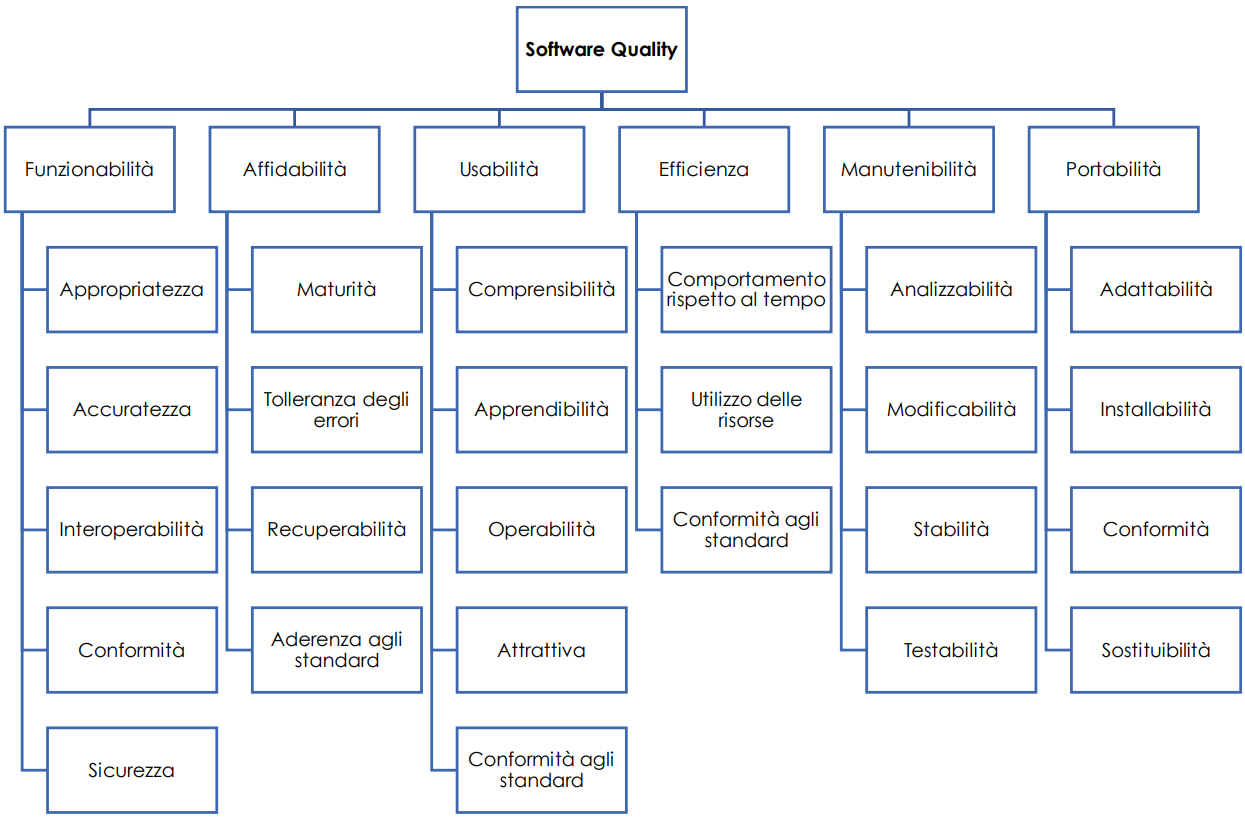
Fare riferimento al documento C16\_SPMP\_Vx.x

1. Standard, metriche e convenzioni
   1. Definizione della qualità

Per la definizione della qualità si fa riferimento allo standard ISO/IEC 9126. Esso definisce sei categorie principali ulteriormente divise in sottocategorie in modo gerarchico. Inoltre, propone delle metriche per la misurazione.

Le norme emesse dallo standard si dividono in 4 parti:

* **Modello della qualità del software:** classificato da sei categorie generali ovvero funzionalità, affidabilità, efficienza, usabilità, manutenibilità, portabilità e per ognuna di esse varie sottocategorie misurabili attraverso delle metriche.
* **Metriche esterne per la qualità:** misurano il comportamento del software sulla base dei test, dall’operatività e dall’osservazione durante la sua esecuzione, in funzione degli obiettivi stabiliti ini un contesto tecnico rilevante o di business.
* **Metriche interne per la qualità:** si applica al software non eseguibile, come il codice sorgente, durante la progettazione e la codifica.
* **Metriche per la qualità in uso:** rappresenta il punto di vista dell’utente sul software. Il livello di qualità in uso è raggiunto quando la qualità interna ed esterna sono raggiunte.

****

Di seguito sono riportati gli attributi considerati fondamentali per la qualità del progetto. Ognuno di esso ha una priorità e un razionale. Gli attributi reputati di scarso interesse non sono riportati.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caratteristica** | **Attributi** | **Priorità** | **Razionale** |
| **Funzionalità** | Appropriatezza | 3 | Mancheranno alcune funzionalità di priorità medio bassa o potrebbero essere simulate (pagamento) |
| Accuratezza | 5 |  |
| Interoperabilità | 4 | Vi è l’interazione con il sistema esterno di pagamento e il dispositivo IoT. |
| Conformità agli standard | 1 | Il processo di sviluppo non richiede l’adozione di standard specifici. |
| **Affidabilità** | Maturità | 3 | Il budget previsto non permette di raggiungere un livello adeguato per questa caratteristica. |
| Tolleranza degli errori | 4 | Le dimensioni del software permettono una gestione adeguata del problema |
| Recuperabilità | 2 | La perdita dei dati non crea particolari problemi, l’ARPA rilascia dei bollettini sulle misurazioni periodicamente e non sono gestiti dalla nostra piattaforma. |
| Conformità agli standard | 2 | Non sono presenti standard di affidabilità ai quali aderire. |
| **Usabilità** | Comprensibilità | 4 | Il software deve risultare molto semplice data la natura del progetto. |
| Apprendibilità | 4 | Risulta importante avere delle interfacce semplici, che risultino familiari e che non necessitano di training. |
| Operabilità | 4 |  |
| Attrattività | 3 | Dato che lo scopo del sistema, oltre che essere funzionale, è finalizzato ad attrarre gli utenti per una giusta causa, la capacità attrattiva deve essere medio-alta. |
| Conformità degli standard | 1 | Non verranno applicati particolari standard. |
| **Efficienza** | Comportamento rispetto al tempo | 3 | La priorità è media in quanto i tempi di attesa non recano particolari problemi. |
| Utilizzo delle risorse | 2 | Risulta una priorità bassa in quanto è presente un extra budget che potrebbe coprire spese per allocare ulteriore memoria. |
| Conformità degli standard | 1 | Non verranno applicati particolari standard. |
| **Manutenibilità** | Analizzabilità | 1 | Non è necessario un sistema di diagnosi in caso di errore |
| Modificabilità | 1 | Come da vincoli del SOW, il software deve essere modulare. |
| Stabilità | 4 |  |
| Testabilità | 4 | Il software deve essere testabile, come richiesto dallo stesso SOW. |
| **Portabilità** | Adattabilità | 2 | Il software è per sua natura adattabile, per renderlo avviabile basta seguire le pposite linee guida. |
| Installabilità | 2 | Il sistema è facilmente installabile, anche grazie all’uso di Maven. |
| Conformità | 4 | È garantita conformità tra gli artefatti prodotti. |
| Sostituibilità | 2 | Non è un requisito del progetto, lo scopo è fornire una nuova piattaforma. |

* 1. Standard per la documentazione
     1. Standard di revisione

Ad ogni modifica effettuata, deve essere aggiornata la Revision History del documento, inserendo la descrizione della modifica, gli autori e la nuova versione.

* + 1. Standard per i documenti

I documenti seguono un template base fornito dai PM e seguono lo stile dei documenti adottato dal TM.

* + 1. Standard per lo scambio di documenti

I documenti devono essere realizzati nel formato .docx, rispettando lo stile del template. Inoltre, tutti i documenti sono condivisi tra i team member e i PM tramite OneDrive dell’account universitario. Infine, i documenti sono consegnati al Top Manager tramite la piattaforma e-learning del dipartimento di Informatica.

* 1. Standard per gli artefatti

Verranno seguite le checklist fornite dal Top Manager.

* 1. Standard di codifica
     1. Nomi dei file

I file sorgente Java devono avere lo stesso nome della classe principale del file.

I file di configurazione devono rispettare gli standard adottati dal framework o tool utilizzato.

I nomi dei file compilati, prodotti dalla build o generati dall’IDE non devono essere modificati.

La nomenclatura dei file deve rappresentare al meglio il loro contenuto.

* + 1. Struttura dei file sorgente

Sarà rispettato [Sus’s Java Style](https://checkstyle.sourceforge.io/sun_style.html).

* + 1. Formattazione

Sarà rispettato [Sus’s Java Style](https://checkstyle.sourceforge.io/sun_style.html).

* + 1. Dichiarazioni

Tramite una dichiarazione di variabile locale è possibile definire più di una variabile, però ogni dichiarazione di una variabile d’istanza deve definirne solo una.

Quando viene dichiarata una variabile locale deve seguirne l’inizializzazione o nella stessa linea o nella successiva.

* + 1. Nomenclatura

Di seguito si elencano i formati che le componenti software del sistema dovranno seguire:

* Package: lettere solo in lowerCamelCase;
* Classi: lettere solamente in UpperCamelCase;
* Metodi: lettere solo in lowerCamelCase;
* Costanti: lettere solo in CONSTANT\_CASE;
* Variabili: lettere solo in lowerCamelCase;
* Parametri: lettere solo in lowerCamelCase. In particolare, quando usati per metodi setter e costruttori, devono avere lo stesso nome delle variabili d’istanza corrispondenti.
  + 1. Documentazione del codice

La documentazione sarà prodotta tramite [Javadoc](https://www.oracle.com/java/technologies/javase/javadoc-tool.html).

* + 1. Base di dati

Ove possibile, le tabelle della base di dati devono rispettare la 3NF (Codd). Tutte le scelte riguardanti la gestione ed il trattamento della base di dati vanno descritti con precisione nella documentazione dedicata.

Le tabelle ed i campi della base di dati seguono i seguenti vincoli di nomenclatura:

* Sono costituiti da sole lettere minuscole;
* Nomi costituiti da più parole separano le stesse tramite underscore (\_);
* Ogni nome deve essere un sostantivo singolare tratto dal dominio del problema ed esplicativo del contenuto.
  1. Standard e pratiche per il testing

Per il testing si fa riferimento al file C16\_TP\_Vx.y.

* 1. Metriche per la valutazione
     1. Metriche di valutazione dei Team Member

Per le valutazioni dei team member si fa riferimento al file C16\_TeamContract\_Vx.y.

* + 1. Metriche di valutazione del codice

**Produzione**

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Descrizione** |
| **BC** | Branch Coverage | Percentuale linee di codice eseguite dai test in seguito ad una esecuzione |
| **LOC** | Lines Of Code | Totale linee di codice |
| **NCLOC** | Non-Comments Lines Of Code | Totale line di codice effettivo |

**Testing**

|  |  |  |
| --- | --- | --- |
| **ID** | **Nome** | **Descrizione** |
| **LOC** | Lines Of Code | Totale linee di codice |
| **NCLOC** | Non-Comments Lines Of Code | Totale line di codice effettivo |

1. Revisione del software

|  |  |  |  |
| --- | --- | --- | --- |
| **Deliverable** | **Data** | **Metodo di Revisione** | **Ulteriori Azioni** |
| **RAD** | 12/12/2022 | Il Top Manager visionerà alcune delle deliverable | Controllo CheckList |
| **SDD** | 12/12/2022 | Il Top Manager visionerà alcune delle deliverable | Controllo Checklist |
| **TCS** | 12/12/2022 | Il Top Manager visionerà alcune delle deliverable |  |
| **ODD** | 17/01/2023 | Il Top Manager visionerà alcune delle deliverable | Controllo Checklist |
| **Sistema Funzionante** | 17/01/2023 | Il Top Manager visionerà alcune delle deliverable |  |

1. Test

Per il testing si fa riferimento al file C16\_TP\_Vx.y.

1. Rapporti sui problemi e azioni correttive

Durante la fase di sviluppo i membri del team saranno controllati dai PM che dovranno accertarsi che i requisiti qualitativi delle deliverable da produrre siano sempre rispettati.

Eventuali problemi saranno discussi tramite i canali di comunicazioni formali e informali messi a disposizione per la realizzazione del progetto.

1. Strumenti, tecniche e metodologie

Per garantire che i criteri di qualità siano rispettati verranno usati i seguenti strumenti:

* Template per la documentazione;
* Linee guida per la scrittura del codice;
* Checklist del Top Manager per il controllo della qualità.

1. Controllo dei dati multimediali

Tutti i media che vengono scambiati tra coloro che lavorano al progetto sono memorizzati su OneDrive o su altri canali di comunicazione.

1. Controllo della fornitura

Non è previsto alcun software di terze parti.

1. Collezione e conservazione dei dati

Per la raccolta e la conservazione degli artefatti viene utilizzata una cartella su OneDrive accessibile sia ai team member che ai PM, tramite l’account universitario.

Il codice viene invece gestito tramite il versioning Git e conservato sulla piattaforma GitHub, in particolare è stata creata una repository dedicata al progetto.

1. Training

Per il training si rimanda al documento C16\_SPMP\_Vx.y.

1. Gestione dei rischi

Per la gestione dei rischi si rimanda al documento C16\_RMP\_Vx.y.

1. Glossario ed Acronimi

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| **SOW** | Statement Of Work |
| **RAD** | Requirement Analysis Document |
| **SDD** | System Design Document |
| **TP** | Test Plan |
| **TCS** | Test Case Specification |
| **ODD** | Object Design Document |
| **RMP** | Risk Management Plan |
| **PM** | Project Manager |
| **TM** | Top Manager |
| **SPMP** | Software Project Management Plan |

1. Procedure di aggiornamento del piano

Si fa riferimento al paragrafo 1.2.