



# QuaranTeam

## Norme di Progetto

Gruppo QuaranTeam - Progetto HD Viz

quaranteam2021@gmail.com

### INFORMAZIONI SUL DOCUMENTO

<b>Versione</b>	3.0.0
<b>Approvatore</b>	Gibellato Alice
<b>Redattori</b>	Veronese Luca Consalvo Federico
<b>Verificatori</b>	Mason Damiano Rech Elia
<b>Stato</b>	Approvato
<b>Uso</b>	Interno
<b>Destinato a</b>	Prof. Vardanega Tullio Prof. Cardin Riccardo QuaranTeam

### Descrizione

Questo documento contiene le linee guida per la gestione di tutti i processi istanziati dal gruppo.

## Registro delle modifiche

Versione	Data	Descrizione	Autore	Verificatore
3.0.0	2021-05-07	Approvazione del documento.	Gibellato Alice	
2.0.4	2021-04-15	Modifica §3.5.2.2.	Veronese Luca	Mason Damiano
2.0.4	2021-04-12	Modifica §2.1.2.4.	Consalvo Federico	Mason Damiano
2.0.3	2021-04-10	Modifica §2.2.2.3.	Consalvo Federico	Mason Damiano
2.0.2	2021-03-29	Modifica §3.1.2.2.	Veronese Luca	Rech Elia
2.0.1	2021-03-28	Modifica §4.1.2.5.	Veronese Luca	Rech Elia
2.0.0	2020-02-28	Approvazione del documento.	Rech Elia	
1.0.3	2021-02-15	Modifica §4.	Veronese Luca	Gibellato Alice
1.0.2	2021-02-12	Modifica §3.	Consalvo Federico	Chiarello Federico
1.0.1	2021-02-10	Modifiche a §1 e §2.	Chiarello Federico	Consalvo Federico
1.0.0	2020-12-27	Approvazione del documento.	Sinigaglia Matteo	
0.4.2	2020-12-22	Revisione della coerenza.	Chiarello Federico	Veronese Luca
0.4.1	2020-12-15	Integrazione §4.	Chiarello Federico	Mason Damiano
0.4.0	2020-12-12	Stesura §4.	Chiarello Federico	Mason Damiano
0.3.1	2020-12-11	Integrazione §3.	Consalvo Federico	Sinigaglia Matteo
0.3.0	2020-12-10	Stesura §3.	Gibellato Alice	Sinigaglia Matteo
0.2.1	2020-12-09	Integrazione §2.	Rech Elia	Veronese Luca
0.2.0	2020-12-08	Stesura §1 e §2.	Consalvo Federico, Rech Elia	Chiarello Federico
0.1.0	2020-12-08	Creazione del documento.	Chiarello Federico	Veronese Luca

# Indice

<b>1 Introduzione</b>	<b>1</b>
1.1 Scopo del documento . . . . .	1
1.2 Scopo del prodotto . . . . .	1
1.3 Glossario . . . . .	1
1.4 Riferimenti . . . . .	1
1.4.1 Riferimenti normativi . . . . .	1
1.4.2 Riferimenti informativi . . . . .	1
<b>2 Processi primari</b>	<b>3</b>
2.1 Fornitura . . . . .	3
2.1.1 Descrizione . . . . .	3
2.1.2 Attività . . . . .	3
2.1.2.1 Studio di Fattibilità . . . . .	3
2.1.2.2 Preparazione della risposta . . . . .	3
2.1.2.3 Contrattazione . . . . .	3
2.1.2.4 Piano di Progetto . . . . .	4
2.1.2.5 Piano di Qualifica . . . . .	4
2.1.2.6 Revisione e valutazione . . . . .	4
2.1.2.7 Consegnna e completamento . . . . .	5
2.1.3 Metriche . . . . .	5
2.1.4 Strumenti . . . . .	5
2.2 Sviluppo . . . . .	5
2.2.1 Descrizione . . . . .	5
2.2.2 Attività . . . . .	5
2.2.2.1 Analisi dei Requisiti . . . . .	5
2.2.2.2 Progettazione . . . . .	7
2.2.2.3 Codifica <sub>G</sub> . . . . .	8
2.2.3 Metriche . . . . .	9
2.2.3.1 QP-1 Sviluppo . . . . .	9
2.2.3.2 QC-1 Manutenibilità . . . . .	9
2.2.3.3 QC-2 Usabilità . . . . .	10
2.2.3.4 QC-3 Efficienza . . . . .	11
2.2.3.5 QC-4 Funzionalità . . . . .	11
2.2.4 Strumenti . . . . .	11
<b>3 Processi di supporto</b>	<b>13</b>
3.1 Documentazione . . . . .	13
3.1.1 Descrizione . . . . .	13
3.1.2 Attività . . . . .	13
3.1.2.1 Implementazione del processo . . . . .	13
3.1.2.2 Documentazione richiesta . . . . .	13
3.1.2.3 Sviluppo e design . . . . .	15
3.1.3 Metriche . . . . .	20
3.1.3.1 QC-5: Comprensibilità . . . . .	20
3.1.4 Strumenti . . . . .	20

<b>3.2 Gestione della configurazione . . . . .</b>	<b>21</b>
3.2.1 Descrizione . . . . .	21
3.2.2 Attività . . . . .	21
3.2.2.1 Versionamento <sub>G</sub> . . . . .	21
3.2.2.2 Gestione delle modifiche . . . . .	23
3.2.3 Metriche . . . . .	23
3.2.4 Strumenti . . . . .	23
<b>3.3 Gestione della qualità . . . . .</b>	<b>24</b>
3.3.1 Descrizione . . . . .	24
3.3.2 Attività . . . . .	24
3.3.2.1 Pianificazione . . . . .	24
3.3.2.2 Garanzia di qualità del prodotto . . . . .	24
3.3.2.3 Garanzia di qualità dei processi . . . . .	24
3.3.2.4 Classificazione . . . . .	24
3.3.3 Metriche . . . . .	25
3.3.3.1 QP-2 Gestione della qualità . . . . .	25
3.3.4 Strumenti . . . . .	25
<b>3.4 Gestione dei cambiamenti . . . . .</b>	<b>26</b>
3.4.1 Descrizione . . . . .	26
3.4.2 Attività . . . . .	26
3.4.2.1 Rilevazione del problema . . . . .	26
3.4.2.2 Risoluzione del problema . . . . .	26
3.4.3 Metriche . . . . .	26
3.4.4 Strumenti . . . . .	27
<b>3.5 Verifica . . . . .</b>	<b>27</b>
3.5.1 Descrizione . . . . .	27
3.5.2 Attività . . . . .	27
3.5.2.1 Analisi . . . . .	27
3.5.2.2 Test . . . . .	28
3.5.3 Metriche . . . . .	29
3.5.3.1 QP-3 Verifica . . . . .	29
3.5.4 Strumenti . . . . .	30
<b>3.6 Validazione<sub>G</sub> . . . . .</b>	<b>30</b>
3.6.1 Descrizione . . . . .	30
3.6.2 Attività . . . . .	31
3.6.2.1 Pianificazione . . . . .	31
3.6.2.2 Test . . . . .	31
3.6.3 Metriche . . . . .	31
3.6.4 Strumenti . . . . .	31
<b>4 Processi organizzativi . . . . .</b>	<b>32</b>
<b>4.1 Gestione organizzativa . . . . .</b>	<b>32</b>
4.1.1 Descrizione . . . . .	32
4.1.2 Attività . . . . .	32
4.1.2.1 Gestione dei ruoli di progetto . . . . .	32
4.1.2.2 Gestione delle comunicazioni . . . . .	34
4.1.2.3 Gestione degli incontri . . . . .	34
4.1.2.4 Gestione degli strumenti di coordinamento . . . . .	35

4.1.2.5	Gestione dei rischi . . . . .	36
4.1.3	Metriche . . . . .	37
4.1.3.1	QP-4 Gestione organizzativa . . . . .	37
4.1.4	Strumenti . . . . .	38
4.2	Formazione . . . . .	39
4.2.1	Descrizione . . . . .	39
4.2.2	Attività . . . . .	39
4.2.2.1	Materiale per la formazione . . . . .	39
4.2.3	Metriche . . . . .	40
4.2.4	Strumenti . . . . .	40

## Elenco delle figure

1	Fasi del ciclo di vita del documento . . . . .	13
2	Struttura canonica delle norme di progetto . . . . .	16
3	Procedura per l'utilizzo di Git . . . . .	23
4	Stati di avanzamento di un'attività nell'Issue Tracking System . . . . .	36
5	Procedura adottata per la gestione dei rischi . . . . .	36

## Elenco delle tabelle

2	Metriche di processo relative al processo di sviluppo . . . . .	9
3	Metriche di prodotto per garantire manutenibilità relative al processo di sviluppo . . . . .	10
4	Metriche di prodotto per garantire usabilità relative al processo di sviluppo . . . . .	10
5	Metriche di prodotto per garantire efficienza relative al processo di sviluppo . . . . .	11
6	Metriche di prodotto per garantire funzionalità relative al processo di sviluppo . . . . .	11
7	Metriche di prodotto per garantire comprensione relative al processo di documentazione . . . . .	20
8	Metriche di processo relative al processo di qualità . . . . .	25
9	Metriche di processo relative al processo di verifica . . . . .	30
10	Metriche di processo relative al processo di gestione organizzativa . . . . .	37
11	Tecnologie e linguaggi di programmazione utilizzati . . . . .	39

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di definire le regole base che tutti i membri del team<sub>G</sub> QuaranTeam sono tenuti a seguire durante lo sviluppo del progetto, al fine di garantire un materiale uniforme e conforme alle normative. Le regole stabilite e discusse nel corpo di questo documento fanno riferimento a tutti i documenti e agli strumenti scelti per lo sviluppo del software<sub>G</sub>. Si utilizzerà un approccio incrementale, partendo dalla stesura del documento nei suoi punti principali e andando progressivamente a normare tutte le decisioni concordate dal team<sub>G</sub>. Tutti i membri del team<sub>G</sub> si impegnano a prendere visione del documento e a seguire le regole stabilite favorendo la coesione tra i membri.

## 1.2 Scopo del prodotto

Il capitolo<sub>G</sub> C4 ha per obiettivo la creazione di un'applicazione web chiamata *HD Viz* sviluppata principalmente in HTML<sub>G</sub>, CSS<sub>G</sub>, JavaScript<sub>G</sub> e con l'utilizzo della libreria<sub>G</sub> D3.js<sub>G</sub>. La piattaforma<sub>G</sub> necessita anche di una parte server<sub>G</sub> che supporti il trasporto della parte HTML<sub>G</sub> verso il browser e l'esecuzione di query<sub>G</sub> su un database<sub>G</sub> SQL<sub>G</sub> o NoSQL<sub>G</sub>, realizzabile in Java<sub>G</sub> con server<sub>G</sub> Tomcat<sub>G</sub> o in JavaScript<sub>G</sub> con server<sub>G</sub> Node.js<sub>G</sub>. L'applicazione ha lo scopo di fornire una visualizzazione di dati, il cui contesto è sconosciuto, a supporto della fase esplorativa dell'analisi dei dati. *HD Viz* dovrà essere in grado di rappresentare dati in almeno 15 dimensioni e fornire minimo 4 diversi tipi di visualizzazione.

## 1.3 Glossario

Viene fornito il *Glossario v3.0.0*, una raccolta di tutti i termini con un significato particolare, che vengono definiti e descritti al fine di evitare ambiguità. In tutti i documenti, i termini definiti nel *Glossario v3.0.0* saranno identificati con una G a pedice.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- **Capitolato<sub>G</sub> d'appalto C4 - HD Viz:**  
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C4.pdf>.

### 1.4.2 Riferimenti informativi

- **Standard ISO/IEC 12207:1995:**  
[https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf);
- **ISO 8601:2000:**  
<http://lists.ebxmxml.org/archives/ebxmxml-core/200104/pdf00005.pdf>;
- **Slide del corso di Ingegneria del Software - Analisi dei requisiti:**  
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L07.pdf>;
- **Documentazione L<sup>A</sup>T<sub>E</sub>X<sub>G</sub>:**  
<https://www.latex-project.org/help/documentation/>;

- **Spunti di studio per la Flipped Classroom sul tema della Documentazione:**  
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/FC2.pdf>;
- **Software Documentation, Ian Sommerville:**  
[www.literateprogramming.com/documentation.pdf](http://www.literateprogramming.com/documentation.pdf);

## 2 Processi primari

### 2.1 Fornitura

#### 2.1.1 Descrizione

Il processo di fornitura ha lo scopo di individuare le norme che il gruppo dovrà rispettare per la realizzazione del prodotto. Il fornitore redige un documento chiamato *Studio di Fattibilità*, nel quale vengono indicati e valutati i rischi della richiesta d'appalto. Il gruppo ha l'incarico di dialogare con il proponente<sub>G</sub> *Zucchetti S.p.A.* e con i committenti<sub>G</sub> per discutere le richieste del progetto, redigendo in un *Piano di Progetto* le attività che intende svolgere per la realizzazione del progetto. Le aspettative sono quelle di ottenere un prodotto che soddisfi le richieste del proponente<sub>G</sub>. Per ottenere ciò il gruppo deve rimanere in stretto contatto con quest'ultimo, per definire i requisiti<sub>G</sub> e chiarire eventuali dubbi sullo sviluppo del prodotto.

#### 2.1.2 Attività

##### 2.1.2.1 Studio di Fattibilità

L'obiettivo dello *Studio di Fattibilità* è analizzare in dettaglio ogni capitolato<sub>G</sub> proposto dal proponente<sub>G</sub>. Nel documento vengono evidenziati gli aspetti positivi e negativi di ogni capitolato<sub>G</sub>. Per ogni capitolato<sub>G</sub> vengono presentate le seguenti sezioni:

- **Informazioni generali:** vengono indicati il nome, il proponente<sub>G</sub> e il committente<sub>G</sub>;
- **Descrizione:** viene redatto un breve riassunto sulle caratteristiche del prodotto da realizzare;
- **Obiettivi:** vengono descritte le attività che il gruppo deve svolgere per il raggiungimento dell'obiettivo;
- **Tecnologie utilizzate:** viene stilata una lista delle tecnologie per la realizzazione del progetto;
- **Vincoli del progetto:** vengono elencati i requisiti<sub>G</sub> obbligatori e opzionali;
- **Valutazione finale:** vengono descritte le note positive e negative del capitolato<sub>G</sub> espresse dal gruppo;
- **Esito:** viene fatta una valutazione generale del capitolato<sub>G</sub>, motivando le ragioni per le quali si è deciso di intraprendere o meno la sua realizzazione.

##### 2.1.2.2 Preparazione della risposta

Verrà scritta una lettera di presentazione per candidare il gruppo alla fornitura del prodotto relativo al capitolato<sub>G</sub> scelto.

##### 2.1.2.3 Contrattazione

In seguito ad un preventivo redatto nel *Piano di Progetto* e una definizione dei casi d'uso<sub>G</sub> redatti nell'*Analisi dei Requisiti*, il gruppo stipula il contratto con il committente<sub>G</sub>, impegnandosi a rispettare le sue richieste.

#### 2.1.2.4 Piano di Progetto

Il responsabile, con l'aiuto degli amministratori, redige un *Piano di Progetto* a cui attenersi durante la realizzazione del progetto. Questo documento è caratterizzato dalle seguenti sezioni:

- **Analisi dei rischi:** in questa sezione si analizzano le difficoltà in cui si potrebbe incorrere durante lo svolgimento del progetto e si analizza in quale modo verranno affrontate tali complicazioni;
- **Modello di sviluppo:** si indica il modello di sviluppo che verrà adottato per l'esecuzione e il rilascio del progetto;
- **Pianificazione:** vengono definite le attività, con relative scadenze, che caratterizzano le varie fasi di realizzazione del progetto;
- **Preventivo e Consuntivo:** viene stipulato un preventivo totale per la realizzazione del progetto. Viene rendicontato il risultato del periodo di attività esprimendo un'analisi su quello che è stato preventivato in anticipo;
- **Organigramma:** viene esposta la composizione del gruppo;
- **Attualizzazione dei rischi:** viene presentato un riepilogo dei rischi riscontrati durante lo sviluppo del progetto. Per ciascuno di essi devono essere indicate le contromisure adottate. La sezione deve contenere una valutazione dell'efficacia di quest'ultime.

#### 2.1.2.5 Piano di Qualifica

Questo documento, redatto dai verificatori, viene realizzato per definire le soluzioni da adottare al fine di garantire la qualità del prodotto. Le attività definite nel *Piano di Qualifica* hanno l'obiettivo di guidare ogni membro del gruppo nella realizzazione corretta del prodotto software<sub>G</sub>. Le attività definite sono le seguenti:

- **Qualità di processo:** vengono definiti gli intervalli ottimali e accettabili entro i quali possono variare le metriche di processo;
- **Qualità di prodotto:** vengono definiti gli intervalli ottimali e accettabili entro i quali possono variare le metriche di prodotto;
- **Strategia di testing:** viene definito il modello con cui viene testato il prodotto per garantirne la corretta implementazione, permettendo il soddisfacimento di tutti i requisiti<sub>G</sub> richiesti;
- **Standard di qualità adottati:** vengono identificati gli standard di qualità adottati per lo sviluppo del prodotto;
- **Resoconto delle attività di verifica:** viene riportato un resoconto delle attività che sono state svolte.

#### 2.1.2.6 Revisione e valutazione

Il gruppo dovrà valutare le caratteristiche del prodotto per garantire la conformità con quanto deciso con il proponente<sub>G</sub> e il committente<sub>G</sub>. Durante la realizzazione il gruppo si impegnerà a coordinare le revisioni delle attività svolte.

### 2.1.2.7 Consegnna e completamento

Al termine dello sviluppo del software<sub>G</sub> il gruppo consegna il prodotto, come stipulato nel contratto.

### 2.1.3 Metriche

In questo processo non è stata rilevata nessuna metrica da poter utilizzare.

### 2.1.4 Strumenti

Di seguito sono descritti gli strumenti identificati dal gruppo per il processo di fornitura.

- **Google Sheets:** utilizzato per la realizzazione di aerogrammi, istogrammi, tabelle e grafici;
- **Microsoft Project:** software<sub>G</sub> per effettuare lo scheduling delle attività, utilizzato per la modellazione dei diagrammi di Gantt.

## 2.2 Sviluppo

### 2.2.1 Descrizione

Il processo di sviluppo racchiude le attività da svolgere per la realizzazione del prodotto software<sub>G</sub> richiesto dal proponente<sub>G</sub>. L'obiettivo di questo processo è il soddisfacimento delle aspettative e delle richieste, rispettando i vincoli tecnici e di design. Il processo mira allo sviluppo di un prodotto software<sub>G</sub> che superi i test e soddisfi i requisiti<sub>G</sub> stabiliti con il proponente<sub>G</sub>.

### 2.2.2 Attività

#### 2.2.2.1 Analisi dei Requisiti

##### Scopo

L'*Analisi dei Requisiti* viene redatta dagli analisti con l'obiettivo di analizzare nello specifico il prodotto da sviluppare e ha i seguenti obiettivi:

- definire lo scopo del lavoro;
- fissare i requisiti concordati con il proponente<sub>G</sub>;
- fornire ai verificatori dei riferimenti per l'attività di controllo.

##### Casi d'uso

Viene adottata la seguente rappresentazione per i casi d'uso:

- **Identificatore:**

UC[codice\_padre].[codice\_figlio]

- **codice\_padre:** indica il caso univoco;
- **codice\_figlio:** indica il sottocaso;
- **Attori primari:** tutti gli attori primari coinvolti;
- **Attori secondari:** tutti gli attori secondari, non obbligatori;

- **Precondizione:** indica lo stato in cui si deve trovare il sistema prima del caso d'uso<sub>G</sub>;
- **Postcondizione:** indica lo stato in cui si deve trovare il sistema dopo il caso d'uso<sub>G</sub>;
- **Scenario principale:** vengono numerati tutti gli eventi che avvengono durante l'esecuzione dell'applicazione;
- **Estensioni:** opzionale, durante l'esecuzione dello scenario principale avviene una condizione che fa deviare il flusso delle attività;
- **Inclusioni:** opzionale, il secondo caso d'uso<sub>G</sub> è incondizionatamente incluso nel primo.

### Classificazione dei requisiti

Viene adottata la seguente nomenclatura per i requisiti<sub>G</sub>:

**R[Tipologia][Importanza][Identificativo]**

Dove:

- **Tipologia**
  - **F:** funzionale, definisce la funzione di un sistema;
  - **Q:** qualitativo, definisce un requisito<sub>G</sub> per garantire la qualità;
  - **P:** prestazionale, definisce un requisito<sub>G</sub> per garantire una buona efficienza del prodotto;
  - **V:** vincolo, definisce un requisito<sub>G</sub> per garantire il vincolo.
- **Importanza:**
  - **O:** requisito<sub>G</sub> obbligatorio;
  - **D:** requisito<sub>G</sub> desiderabile;
  - **F:** requisito<sub>G</sub> facoltativo.
- **Identificativo:** l'identificativo è univoco e viene rappresentato da un numero separato da un punto: l'identificativo è in formato [padre.figlio].

La sezione §4 del documento *Analisi dei Requisiti v3.0.0* è dedicata alla classificazione dei requisiti<sub>G</sub> e riporta per ogni requisito<sub>G</sub>:

- **Descrizione:** viene descritto in modo breve e conciso il requisito<sub>G</sub>;
- **Fonti:** vengono suddivise le varie fonti in base alla provenienza dei requisiti<sub>G</sub>. Queste si suddividono in:
  - **Capitolato<sub>G</sub>:** riporta un requisito<sub>G</sub> individuato nel capitolato<sub>G</sub>;
  - **Interno:** l'aggiunta del requisito<sub>G</sub> ritenuta opportuna dagli analisti;
  - **Casi d'uso<sub>G</sub>:** il requisito<sub>G</sub> viene individuato all'interno in uno o più casi d'uso.
  - **Verbale:** dopo un chiarimento con il proponente<sub>G</sub> è possibile che venga individuato un nuovo requisito<sub>G</sub>, che verrà riportato nei verbali.

### 2.2.2.2 Progettazione

#### Scopo

La progettazione precede la codifica<sub>G</sub> perseguiendo la correttezza per costruzione e serve a dominare la complessità del prodotto.

#### Descrizione

La progettazione definisce le caratteristiche del prodotto software<sub>G</sub> richiesto in funzione dei requisiti<sub>G</sub> specificati dall'*Analisi dei Requisiti*. In questa fase i progettisti hanno il compito di definire una possibile soluzione del problema che sia soddisfacente per tutti gli stakeholder<sub>G</sub>.

#### Qualità dell'architettura

L'architettura logica del prodotto dovrà godere delle seguenti qualità:

- **sufficienza:** capace di soddisfare tutti i requisiti<sub>G</sub>;
- **modularità:** suddivisa in parti chiare e ben distinte;
- **robustezza:** capace di sopportare ingressi diversi da parte dell'utente e dell'ambiente, non facendo assunzioni ottimistiche;
- **efficienza:** in termini di tempo, di spazio e nelle comunicazioni;
- **affidabilità:** garantisce di ottenere il risultato desiderato se utilizzata correttamente;
- **semplicità:** ogni parte che la compone contiene solamente il necessario e nulla di superfluo;
- **incapsulamento:** l'interno delle componenti non è visibile dall'esterno, facendo ricorso al principio dell'information hiding<sub>G</sub>.

#### Qualità della progettazione

Per perseguire la qualità nella progettazione dell'architettura è necessario seguire le seguenti regole:

- assegnare nomi significativi e parlanti a package, classi, metodi e variabili;
- quando possibile, prediligere sempre l'utilizzo di opportuni design pattern<sub>G</sub>;
- evitare l'utilizzo dell'ereditarietà tra classi concrete, prediligendo sempre l'uso di classi astratte e/o interfacce;
- perseguire sempre il principio dell'incapsulamento<sub>G</sub> e dell'information hiding<sub>G</sub>;
- evitare la definizione di dipendenze circolari tra classi.

#### Technology Baseline<sub>G</sub>

La Technology Baseline<sub>G</sub> è redatta dai progettisti e deve includere:

- **Proof of Concept<sub>G</sub>:** primo eseguibile del sistema con funzione dimostrativa;
- **tecniche utilizzate:** descrizione delle tecnologie impiegate nello sviluppo del progetto che include gli aspetti positivi o negativi riscontrati;
- **test di integrazione:** definizione dei test eseguiti per verificare che le componenti del sistema, una volta integrate insieme, interagiscano in modo corretto;

## Product Baseline<sub>G</sub>

La Product Baseline<sub>G</sub> è redatta dai progettisti e deve includere:

- **diagrammi UML<sub>G</sub>:**
  - diagrammi delle classi;
  - diagrammi dei package;
  - diagrammi di sequenza;
- **definizione delle classi:** ogni classe deve essere descritta in modo da spiegarne in maniera esaustiva lo scopo e le funzionalità;
- **design pattern<sub>G</sub>:** descrizione dei design pattern<sub>G</sub> utilizzati per realizzare l'architettura. Ogni design pattern<sub>G</sub> deve essere accompagnato da una descrizione e da un diagramma che ne mostri la struttura;
- **test di unità<sub>G</sub>:** definizione dei test eseguiti per verificare il corretto funzionamento individuale delle classi e dei metodi che implementano il sistema software<sub>G</sub>;
- **tracciamento delle classi:** associazione tra requisiti<sub>G</sub> e classi che li soddisfano.

### 2.2.2.3 Codifica<sub>G</sub>

#### Scopo

La fase di codifica<sub>G</sub> ha lo scopo di normare la realizzazione del prodotto software<sub>G</sub>. I programmatore devono attenersi alle norme elencate durante la fase di programmazione. L'uso di convenzioni e norme di programmazione permette di generare codice leggibile ed uniforme, andando ad agevolare le fasi di manutenzione, verifica e validazione<sub>G</sub> e porta ad un miglioramento della qualità del codice.

#### Descrizione

Il codice dovrà rispettare le norme stabilite in questo documento, con il fine di perseguire un buon livello di qualità.

#### Stile di codifica<sub>G</sub>

Per garantire uniformità nel codice del progetto è necessario rispettare le seguenti norme:

- **parentesizzazione:** le parentesi graffe verranno utilizzate sempre, anche se il corpo del blocco è vuoto o presenta una sola istruzione;
- **indentazione:** i blocchi del codice devono essere correttamente indentati. L'indentazione dei commenti non viene considerata;
- **scrittura dei metodi:** è desiderabile mantenere i metodi brevi, limitando il numero di righe di codice;
- **univocità dei nomi:** classi, metodi e variabili devono avere nomi univoci e significativi;
- **classi:** i nomi delle classi devono iniziare sempre con la lettera maiuscola, attenendosi all'UpperCamelCase<sub>G</sub>;
- **metodi e variabili:** i nomi di metodi e variabili devono iniziare sempre con la lettera minuscola, attenendosi al LowerCamelCase<sub>G</sub>;

- **costanti:** i nomi delle costanti devono essere scritti usando solo maiuscole;
- **ricorsione:** l'uso della ricorsione va evitato quando possibile, in quanto aumenta la complessità computazionale del codice.

## JavaScript<sub>G</sub>

In questa sezione vengono riportate delle linee guida riguardanti il linguaggio JavaScript<sub>G</sub> a cui attenersi nella realizzazione del prodotto:

- Prediligere l'utilizzo di **const** e **let** rispetto a **var** per la dichiarazione delle variabili;
- Prediligere la notazione "a freccia" ( $=>$ ) per le funzioni anonime;
- Attenersi alle convenzioni stabilite da CommonJs<sub>G</sub>;
- Favorire il disaccoppiamento, sviluppando moduli che racchiudono funzionalità affini;
- Limitare la dichiarazione di variabili nello scope globale.

### 2.2.3 Metriche

#### 2.2.3.1 QP-1 Sviluppo

Tabella 2: Metriche di processo relative al processo di sviluppo

Codice	Nome	Descrizione	Formula
QM-PROC-1	Copertura requisiti obbligatori (RO)	Indice che misura la percentuale di requisiti obbligatori soddisfatti.	$\frac{\#RO \ coperti}{\#totale \ RO} \times 100$ <p>Dove:</p> <ul style="list-style-type: none"> <li>• RO = requisiti obbligatori.</li> </ul>
QM-PROC-2	Copertura requisiti non obbligatori (RNO)	Indice che misura la percentuale di requisiti desiderabili e facoltativi soddisfatti.	$\frac{\#RD \ e \ RF \ coperti}{\#totale \ RD \ e \ RF} \times 100$ <p>Dove:</p> <ul style="list-style-type: none"> <li>• RD = requisiti desiderabili;</li> <li>• RF = requisiti facoltativi.</li> </ul>

#### 2.2.3.2 QC-1 Manutenibilità

La manutenibilità rappresenta una delle caratteristiche principali del modello di qualità stabilito dallo standard di riferimento. Tale caratteristica vuole raccogliere l'insieme di attributi riguardanti lo sforzo richiesto per apportare modifiche specifiche al prodotto.

Tabella 3: Metriche di prodotto per garantire manutenibilità relative al processo di sviluppo

Codice	Nome	Descrizione	Formula
QM-PROD-1	Complessità ciclomatica (CC)	Indice che stima la complessità di un programma attraverso l'analisi del codice sorgente.	$V(G) = E - N + P$ <p>Dove:</p> <ul style="list-style-type: none"> <li>• E = # archi;</li> <li>• N = # nodi;</li> <li>• P = # componenti connesse.</li> </ul>
QM-PROD-2	Densità di duplicazione (DD)	Percentuale di linee di codice ripetute nei file sorgenti.	$\frac{\# \text{linee di codice ripetute}}{\# \text{linee totali di codice}} \times 100$
QM-PROD-3	Numero di bug (NB)	Numero di bug rilevati nei file sorgenti.	
QM-PROD-4	Numero di code smell (NCS)	Numero di code smell rilevati nei file sorgenti.	
QM-PROD-5	Comprensione del codice (CDC)	Percentuale di comprensione del codice, calcolata tramite le linee di codice e linee di commenti	$\frac{\# \text{linee di codice}}{\# \text{linee di commenti}} \times 100$

#### 2.2.3.3 QC-2 Usabilità

L'usabilità rappresenta una delle caratteristiche principali del modello di qualità stabilito dallo standard di riferimento. Tale caratteristica vuole raccogliere l'insieme di attributi riguardanti lo sforzo necessario all'utilizzo del prodotto e la valutazione individuale su tale uso da parte di un insieme di utenti.

Tabella 4: Metriche di prodotto per garantire usabilità relative al processo di sviluppo

Codice	Nome	Descrizione	Formula
QM-PROD-6	Click necessari (CN)	Valore che indica i click necessari per raggiungere la funzionalità richiesta dall'utente.	

### 2.2.3.4 QC-3 Efficienza

L'efficienza rappresenta una delle caratteristiche principali del modello di qualità stabilito dallo standard di riferimento. Tale caratteristica vuole raccogliere l'insieme di attributi riguardanti il rapporto tra il livello delle prestazioni e la quantità di risorse usate durante la loro esecuzione, sotto condizioni prestabilite.

Tabella 5: Metriche di prodotto per garantire efficienza relative al processo di sviluppo

Codice	Nome	Descrizione	Formula
QM-PROD-7	Tempo medio di risposta (TMR)	Tempo medio impiegato dal software per rispondere ad una certa richiesta dell'utente o svolgere un'attività di sistema.	

### 2.2.3.5 QC-4 Funzionalità

La funzionalità rappresenta una delle caratteristiche principali del modello di qualità stabilito dallo standard di riferimento. Tale caratteristica vuole raccogliere l'insieme di attributi riguardanti un insieme di funzioni e le loro proprietà. Tali funzioni mirano a soddisfare requisiti<sub>G</sub> stabiliti o implicitamente dedotti.

Tabella 6: Metriche di prodotto per garantire funzionalità relative al processo di sviluppo

Codice	Nome	Descrizione	Formula
QM-PROD-8	Errori di utilizzo (EDU)	Errori inaspettati nell'utilizzo del software non segnalati dal sistema.	

### 2.2.4 Strumenti

Di seguito sono descritti gli strumenti identificati dal gruppo per il processo di sviluppo.

- **StarUML:** software per la modellazione di diagrammi UML<sub>G</sub>;
- **IntelliJ IDEA:** IDE utilizzato per lo sviluppo di applicativi JavaScript<sub>G</sub>;
- **Visual Studio Code:** IDE utilizzato per lo sviluppo di codice JavaScript<sub>G</sub>, HTML<sub>G</sub> e CSS<sub>G</sub>;
- **Node.js<sub>G</sub>:** utilizzato per la parte server di supporto alla presentazione nel browser e al database e per l'esecuzione di codice JavaScript;
- **Experss.js<sub>G</sub>:** framework<sub>G</sub> per applicazioni web Node.js<sub>G</sub>;

- **ESLint<sub>G</sub>**: strumento utilizzato per effettuare l'analisi statica del codice JavaScript<sub>G</sub>;
- **Mocha<sub>G</sub>**: framework<sub>G</sub> di test JavaScript<sub>G</sub> eseguito su Node.js<sub>G</sub> e su browser per lo svolgimento di test asincroni;
- **SonarQube<sub>G</sub>**: piattaforma usata per effettuare analisi statica del codice. Utilizzata per la rilevazione delle metriche legate alla codifica<sub>G</sub>.

### 3 Processi di supporto

#### 3.1 Documentazione

##### 3.1.1 Descrizione

Il processo di documentazione ha lo scopo di raccogliere e organizzare in documenti le informazioni prodotte all'interno di un processo o di un'attività. Deve prevedere un insieme di attività mirate a supportare la produzione, la verifica e la manutenzione della documentazione. Questa sezione contiene le decisioni e le norme scelte per la stesura, la verifica e l'approvazione della documentazione ufficiale. Si vogliono stabilire norme, linee guida e fasi del processo che consentano di produrre una documentazione che soddisfi le metriche per la qualità definite in questo documento. Tali norme sono tassative per ogni documento formale prodotto.

##### 3.1.2 Attività

###### 3.1.2.1 Implementazione del processo

###### Ciclo di vita del documento

Ogni documento attraversa le fasi del seguente ciclo di vita:

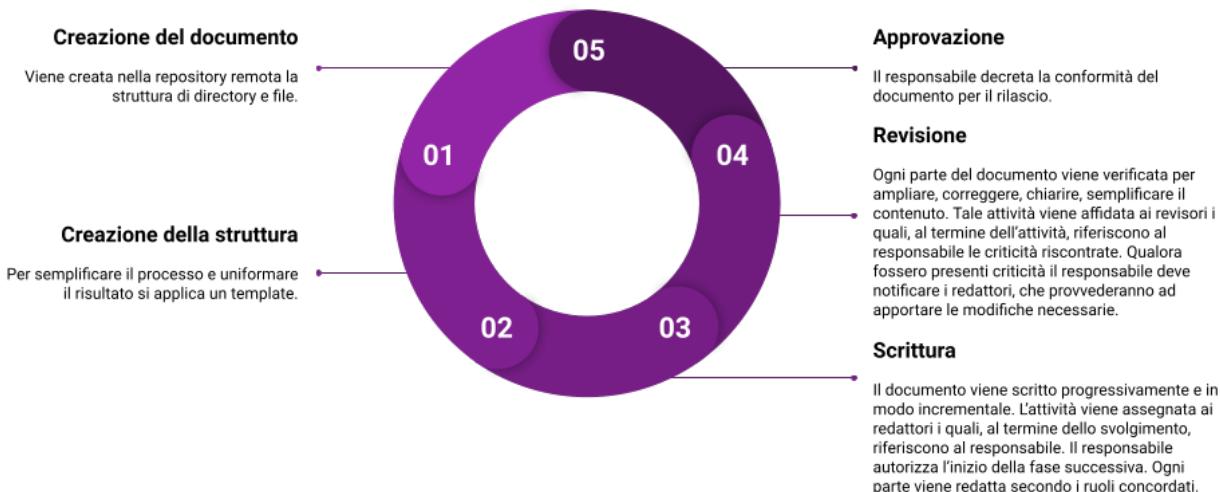


Figura 1: Fasi del ciclo di vita del documento

###### 3.1.2.2 Documentazione richiesta

###### Documenti interni

I documenti interni sono:

- **Studio di Fattibilità - SdF;**
- **Norme di Progetto - NdP.**

## Documenti esterni

I documenti esterni sono:

- **Analisi dei Requisiti - AdR;**
- **Piano di Progetto - PdP;**
- **Piano di Qualifica - PdQ;**
- **Glossario - G;**
- **Manuale Utente - MU:** manuale ad uso degli utilizzatori del software<sub>G</sub>;
- **Manuale del Manutentore - MM:** manuale rivolto agli sviluppatori e ai manutentori del software<sub>G</sub>.

## Verbali

I verbali riassumono gli argomenti discussi nelle riunioni e riepilogano le decisioni prese in esse. Non sono da confondere con i resoconti, i quali non vengono verbalizzati. Nei verbali viene indicata una tabella con le decisioni intraprese dal gruppo che abbiano impatto su azioni future. Seguono la medesima struttura degli altri documenti, e in aggiunta contengono obbligatoriamente le seguenti sezioni:

- **Informazioni generali:** sezione composta dai seguenti punti:
  - luogo della riunione, sia esso fisico o virtuale;
  - data della riunione;
  - ora di inizio della riunione;
  - ora di fine della riunione;
  - elenco dei membri del gruppo presenti.

Qualora fosse assente almeno un membro deve essere riportata in questa sezione anche la lista dei membri del gruppo assenti.

- **Ordine del giorno:** lista degli argomenti che il gruppo intende discutere nel corso della riunione.
- **Resoconto:** sintesi della discussione svolta intorno ai punti dell'ordine del giorno;
- **Riepilogo delle decisioni:** tabella contenente la lista delle decisioni e il relativo codice identificativo.

Anche i verbali si classificano in interni o esterni, in base alla loro destinazione.

## Manuali

I manuali espongono in maniera esauriente le informazioni necessarie all'utente più o meno esperto dell'applicazione (Manuale Utente) o al manutentore (Manuale del Manutentore).

Il Manuale Utente deve essere focalizzato sui casi d'uso di utilizzo dell'applicazione. Questi devono essere illustrati in un'apposita sezione, mediante elenchi numerati e corredati con immagini delle componenti dell'interfaccia utente, con didascalia. Il documento deve contenere inoltre una sezione

che illustra nel dettaglio le componenti e le funzionalità messe a disposizione dall'applicazione, una guida all'installazione e le istruzioni per contattare il supporto tecnico.

Il Manuale del Manutentore fornisce una guida agli sviluppatori interessati ad estendere o mantenere l'applicazione. Esso deve contenere:

- un'introduzione all'architettura realizzata, con focus sui design pattern<sub>G</sub> noti utilizzati;
- una descrizione delle scelte progettuali e delle tecnologie adottate.

La descrizione dell'architettura e dei design pattern adottati deve fare uso di diagrammi UML 2.x, corredati da descrizione testuale.

Ogni manuale deve contenere inoltre un glossario proprio.

### 3.1.2.3 Sviluppo e design

#### Convenzione sul formato

Il formato di esportazione è il PDF, non necessariamente autenticato (PDF/A). Ogni documento ha gli attributi *autore* e *creatore* valorizzati con il nome del gruppo e l'attributo *nome* valorizzato con il titolo del documento. Se composto da più parole queste devono essere separate da un carattere "blank", e solo i sostantivi e la prima parola devono iniziare con la lettera maiuscola (es. Norme di Progetto).

#### Convenzione sulla nomenclatura

##### Documenti

I nomi dei file coincidono con la conversione in LowerCamelCase<sub>G</sub> del nome del documento, senza omissione delle preposizioni. Ad essi viene riportata anche la versione corrente preceduta dal simbolo '\_'. I seguenti sono esempi **corretti** sono:

- pianoDiProgetto\_v1.0.0.pdf;
- pianoDiQualifica\_v3.0.0.pdf.

Alcuni esempi **non corretti** sono:

- piano\_Di\_Progetto.pdf (usa un carattere separatore);
- PianoDiQualifica.pdf (inizia con lettera maiuscola);
- normeProgetto.pdf (omette la preposizione);
- analisiDeiRequisiti-v1.0.0.pdf(utilizza il simbolo '-' invece di '\_').

##### Verbali

Per i nomi dei verbali è stata adottata una speciale nomenclatura. Tale scelta permette di identificare univocamente ogni verbale. Ciascun verbale presenta quindi la seguente forma:

**V.[X].[YYYY-MM-DD]**

Dove:

- V: indica che si tratta di un verbale;
- X può essere:

- I: se l'incontro verbalizzato è un incontro interno;
- E: se l'incontro verbalizzato è un incontro esterno.
- YYYY-MM-DD: indica la data in cui l'incontro verbalizzato si è svolto.

Ogni decisione presa in una riunione riporta il codice V[X]\_[YYYY-MM-DD].[decisione], dove *decisione* è un numero intero unico per ogni decisione all'interno dello stesso verbale, e i rimanenti parametri assumono i significati descritti sopra. Il nome di un file .pdf relativo ad un verbale è della forma V[X]\_[YYYY-MM-DD].pdf.

## Uniformità dei documenti

Il gruppo si impegna a mantenere una certa uniformità nella stesura dei documenti. Ogni qualvolta viene fatto riferimento a una sezione del documento stesso o di un altro documento, questo riferimento viene indicato con il simbolo '§' seguito dal numero della sezione e dal titolo della sezione scritto tra parentesi. I contenuti di ciascun documento vengono indicati nelle norme di progetto, dove viene seguita una struttura ben precisa.

## Norme di progetto

La struttura canonica del documento è:

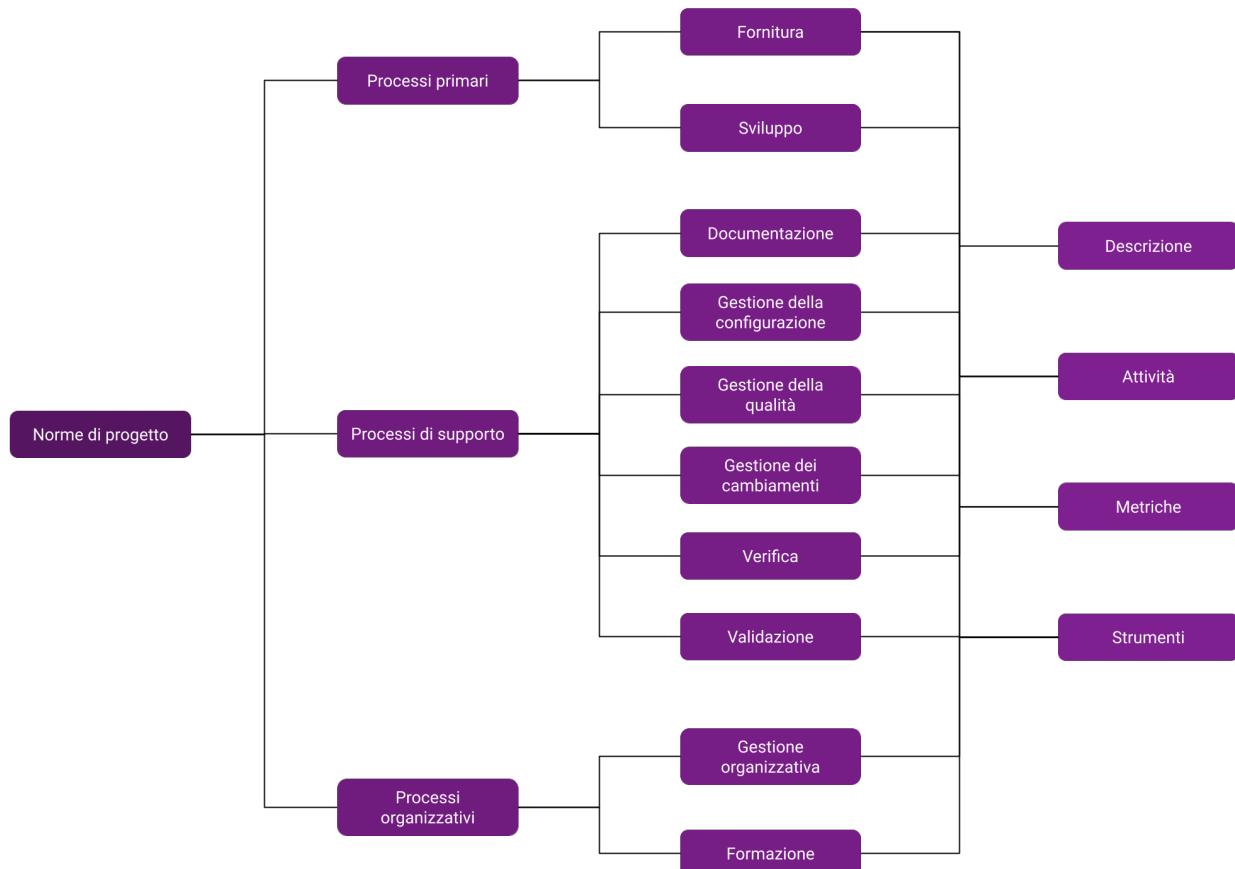


Figura 2: Struttura canonica delle norme di progetto

## Altri documenti

Per quanto riguarda gli altri documenti non si segue una struttura rigorosa a livello di organizzazione dei contenuti: i nomi delle sezioni, la presentazione e la disposizione dei contenuti vengono decisi durante gli incontri interni secondo necessità. L'inserimento di nuove sezioni viene proposto ai membri e approvato dal responsabile.

## Parti comuni

Tra i diversi documenti che vengono richiesti vi sono delle parti comuni.

### Frontespizio

Il frontespizio è la prima pagina comune a ogni documento ed è così strutturato:

- **Logo del gruppo:** logo di *QuaranTeam*, centrato e visibile come elemento più in alto;
- **Titolo:** nome del documento, centrato e visibile appena sotto il logo;
- **Gruppo e progetto:** nome del gruppo e del progetto *HD Viz*, centrato e visibile appena sotto il titolo;
- **E-mail del gruppo:** indirizzo e-mail del gruppo, centrato e visibile appena sotto i nomi del gruppo e del progetto;
- **Tabella delle informazioni:** situata al centro della pagina, riporta l'intestazione INFORMAZIONI SUL DOCUMENTO. Le righe riportano le seguenti informazioni:
  - **Versione:** versione del documento;
  - **Approvatore:** nominativo del membro del gruppo incaricato all'approvazione del documento;
  - **Redattori:** nominativi dei membri del gruppo incaricati della redazione del documento;
  - **Verificatori:** nominativi dei membri del gruppo incaricati della verifica del documento;
  - **Stato:** stadio corrente del ciclo di vita del documento;
  - **Uso:** destinazione d'uso del documento, "interno" o "esterno";
  - **Destinatari:** destinatari del documento.
- **Descrizione:** breve descrizione relativa al documento (massimo 4 righe), centrata e situata nella parte inferiore della pagina.

### Registro delle modifiche

Il registro delle modifiche inizia nella seconda pagina e può occupare una o più pagine. Consiste in una tabella le cui colonne contengono nell'ordine i seguenti valori:

- versione del documento a seguito della modifica;
- data della modifica;
- descrizione della modifica che comprende il verificatore;
- nominativo dei membri che hanno portato a termine la modifica;
- ruoli dei membri che hanno portato a termine la modifica.

Le righe della tabella devono avere tutti i campi valorizzati e sono ordinate in ordine temporale dal più recente al meno recente.

## Indice

L'indice ha il fine di fornire la visione macroscopica della struttura del documento. Esso inizia nella pagina che segue l'ultima pagina del registro delle modifiche. Se sono presenti tabelle o immagini all'interno del documento si può includere una sezione per indicizzare le tabelle e/o le immagini.

## Contenuto principale

Il contenuto principale è suddiviso in sezioni, sottosezioni, sotto-sottosezioni, paragrafi e sottoparagrafi. La loro redazione deve seguire le norme tipografiche riportate nella sezione §3.1.3.5.

## Impostazione della pagina

La struttura di ogni pagina ad eccezione della prima è la seguente:

- in alto a sinistra è presente il nome del documento, seguito dalla versione;
- in alto a destra è presente il logo del gruppo;
- il contenuto è posto tra l'intestazione e il piè di pagina, ed è delimitato superiormente e inferiormente da linee orizzontali;
- in basso a sinistra è riportato il numero dell'ultimo sottoparagrafo contenuto nella pagina (non presente nei verbali e nel glossario);
- in basso a destra è riportato il numero di pagina corrente insieme al numero di pagine totali del documento.

## Template

Per garantire l'adesione del documento alle regole espresse in questa sezione sono stati definiti:

- un template per i verbali;
- un template per le lettere di presentazione;
- un template per i documenti di altro tipo.

È stata definita all'interno della directory Utility la classe stdDocument.cls, che contiene comandi utili all'impostazione della pagina e alla generazione del frontespizio, del registro delle modifiche e dell'indice per le tre tipologie di documenti. Il codice sorgente di un documento deve iniziare con il comando documentclass{stdDocument}. Per inserire le informazioni corrette nella prima pagina di ogni documento è necessario inoltre generare per ogni documento una copia del file command.tex contenuto nella directory Utility e ridefinirne gli opportuni comandi. L'applicazione del template permette altresì di ridurre il tempo necessario alla stesura dei documenti e di svincolare i redattori da un lavoro di tipografia avanzata.

## Norme tipografiche e redazionali

### Riferimenti

Tutti i riferimenti normativi ed informativi devono essere inseriti in un elenco puntato al termine della sezione di introduzione, qualora fossero necessari. Il nome di un documento deve essere riportato per intero. Ogni riferimento a un documento richiesto dal regolamento del progetto di Ingegneria del Software deve riportare il titolo del documento, la versione, il numero e il nome del paragrafo a cui si fa riferimento.

Qualora un libro debba essere inserito tra i riferimenti ne deve essere riportata l'edizione e i capitoli di riferimento.

## Glossario

Ogni termine la cui definizione è presente nel *Glossario* viene contrassegnato da un carattere "G" in pedice e in corsivo al termine della parola.

## Date e orari

La rappresentazione delle date deve seguire il formato **YYYY-MM-DD** previsto dallo standard ISO 8601:2000, dove:

- ogni carattere Y si riferisce a una e una sola cifra dell'anno;
- ogni carattere M si riferisce a una e una sola cifra del mese;
- ogni carattere D si riferisce a una e una sola cifra del giorno.

La rappresentazione degli orari deve seguire il formato hh:mm, in cui esattamente due cifre sono utilizzate per indicare l'ora (compresa tra 0 e 23) ed esattamente due cifre sono utilizzate per indicare il minuto. L'ora è da intendersi secondo il fuso orario dell'Europa Centrale, ossia con un offset positivo di 1 ora rispetto all'orario UTC durante il periodo di ora solare e un offset positivo di 2 ore durante il periodo di ora legale.

## Elenchi puntati e numerati

Ogni voce di un elenco ordinato o numerato che non definisce dei termini comincia con la lettera minuscola e termina con un carattere punto e virgola (';'), tranne per l'ultimo elemento dell'elenco che termina con un punto fermo ('.'). Ogni voce di un elenco che definisce dei termini deve iniziare con il termine da definire, in grassetto.

## Stile del testo

- **grassetto:** viene applicato se necessario alle voci di un elenco di definizioni o a termini sui quali si vuol far ricadere l'attenzione del lettore;
- **corsivo:** vengono scritti in corsivo il nome del proponente<sub>G</sub>, il nome del gruppo, il nome del progetto e i nomi dei documenti. Per quanto riguarda i nomi dei documenti se si fa riferimento ad un documento di *QuaranTeam* si aggiunge la versione del documento (ad es. *Analisi dei Requisiti v3.0.0*), altrimenti il nome viene riportato in corsivo senza la versione;
- **punteggiatura:** deve essere utilizzata per isolare frasi coese e coerenti. Inoltre:
  - le parentesi aperte e nessun altro segno di punteggiatura devono essere precedute da spazi;
  - gli apici singoli ' ' devono essere utilizzati per racchiudere tra di essi un singolo carattere o segno, gli apici doppi " " possono racchiudere una o più parole;
  - il simbolo '/' non può essere utilizzato in sostituzione di una congiunzione, ma può essere utilizzato nella disgiunzione non esclusiva "e/o".

## Elementi grafici

Deve essere riportata la didascalia numerata per tavole, immagini e diagrammi UML<sub>G</sub>.

Per una migliore leggibilità il colore di sfondo di una riga di una tabella è diverso da quello della riga inferiore e da quella superiore. Se la tabella si estende su più pagine deve essere riportata la didascalia in ogni pagina.

## Stile redazionale

Deve essere privilegiato l'uso di verbi in modo attivo e personale, in modo tale da rendere sempre chiaro ed esplicito quale sia il soggetto dell'azione.

La chiarezza espositiva deve essere privilegiata nei confronti della fluidità di lettura.

### 3.1.3 Metriche

#### 3.1.3.1 QC-5: Comprensibilità

La comprensibilità rappresenta una delle sotto-caratteristiche principali dell'usabilità nel modello di qualità stabilito dallo standard di riferimento. Tale caratteristica viene sfruttata per garantire una certa comprensibilità nei documenti che il gruppo produce. Ogni documento deve essere chiaro, leggibile e comprensibile. Queste qualità sono largamente influenzate dalla correttezza ortografica, sintattica e semantica.

Tabella 7: Metriche di prodotto per garantire comprensione relative al processo di documentazione

Codice	Nome	Descrizione	Formula
QM-PROD-9	Indice di GULPEASE (IDG)	È un numero decimale positivo che rappresenta il grado di scolarizzazione necessario a comprendere il documento in analisi.	$\frac{300 \times \text{frasi} - 10 \times \text{lettere}}{\text{parole}}$
QM-PROD-10	Correttezza ortografica (CO)	È un numero intero positivo che indica il numero di errori ortografici presenti nel documento in analisi.	

### 3.1.4 Strumenti

Di seguito sono descritti gli strumenti identificati per il processo di documentazione.

- **TEX<sub>G</sub>**: utilizzato per la stesura dei documenti. Permette il raggiungimento di un'elevata qualità tipografica e il versionamento<sub>G</sub> del sorgente della documentazione.
- **MiKTeX<sub>G</sub>**: utilizzato per la compilazione dei sorgenti.
- **Editor di testo**: vengono utilizzati principalmente TeXstudio<sub>G</sub> e Texmaker<sub>G</sub>. È accettato l'uso di qualsiasi editor di testo.
- **StarUML**: utilizzato per la produzione dei diagrammi UML<sub>G</sub>;
- **diagrams.net<sub>G</sub>**: utilizzato per altri schemi o diagrammi da inserire nella documentazione. Tale strumento facilita la condivisione e la sincronizzazione del lavoro da parte dei membri del gruppo.

## 3.2 Gestione della configurazione

### 3.2.1 Descrizione

Lo scopo del processo di gestione della configurazione è creare ordine tra software<sub>G</sub> e documenti. Tutto ciò che è configurato si trova in un posto preciso, ha uno stato identificativo, è modificato secondo procedure e posto sotto versionamento<sub>G</sub>.

### 3.2.2 Attività

#### 3.2.2.1 Versionamento<sub>G</sub>

##### Codice di versione del documento

Per ogni documento deve essere possibile ricostruire lo storico delle modifiche che ha subito. Per questo ogni modifica provoca la creazione di una nuova versione del documento, che viene tracciata nella tabella delle modifiche, identificabile dal suo numero di versione. Nella descrizione di ogni modifica viene indicato il relativo verificatore. Ogni numero di versione è composto da tre cifre:

X.Y.Z

- X: rappresenta una versione del documento con tutte le caratteristiche desiderabili per un rilascio; il numero:
  - inizia da 0;
  - viene incrementato all'approvazione del documento dal responsabile di progetto, quando è pronto per il rilascio.
- Y: rappresenta una versione che ha modificato la struttura del documento; il numero:
  - inizia da 0;
  - viene incrementato all'aggiunta, spostamento o rimozione di una qualche sezione dall'autore della modifica;
  - viene riportato a 0 all'incremento di X.
- Z: rappresenta una versione ancora in fase di lavorazione, e quindi soggetta a modifiche, del documento; il numero:
  - inizia da 0;
  - viene incrementato dal redattore della modifica del corpo del documento; rientrano in questo caso anche le modifiche apportate durante una revisione o verifica del documento;
  - viene riportato a 0 all'incremento di Y.

##### Organizzazione dei repository<sub>G</sub>

I membri del team<sub>G</sub> possono interagire con il VCS (Version Control System<sub>G</sub>) sia da linea di comando, sia attraverso software<sub>G</sub> che ne migliorano l'usabilità. La versione comune e ufficiale del progetto è ospitata in un repository<sub>G</sub> remoto disponibile all'indirizzo:

<https://github.com/QuaranTeam2021/HD-Viz>

Ci sono due tipi di repository<sub>G</sub>:

- **locale:** ogni membro del gruppo lavora sui file clonati dal repository<sub>G</sub> remoto nel proprio computer;
- **remoto:** pubblicato su GitHub<sub>G</sub>, contiene il lavoro svolto da ogni componente e che viene condiviso con il team<sub>G</sub>.

La struttura è la stessa per entrambe le copie.

Il repository<sub>G</sub> dei documenti è così organizzato:

- **Utility:** contiene tutti i file utili a tutti i comandi di implementazione, tra cui i comandi L<sup>A</sup>T<sub>E</sub>X<sub>G</sub> personalizzati e il template dei documenti;
- **interni:** contiene tutti i documenti destinati ad uso interno del team<sub>G</sub>;
- **esterni:** contiene la documentazione destinata ad uso esterno;
- **immagini:** contiene le immagini comuni a tutti i documenti;
- una cartella per ogni documento, che contiene tutte le risorse necessarie alla sua costruzione.

La suddivisione dei file per revisione aiuta ad evidenziare il lavoro svolto per ogni consegna e migliora la tracciabilità dei file all'interno di ognuna di esse.

### Organizzazione dei branch

Per fare in modo che ogni modifica fatta non sia distruttiva, ognuna verrà effettuata su un determinato branch. La scelta del branch su cui lavorare dovrà essere effettuata seguendo queste direttive:

- il branch master verrà modificato solo ad ogni avanzamento significativo per il prodotto (baseline);
- il branch develop è quello principale: verrà modificato ogni volta che una qualche funzionalità è stata completata e testata;
- a partire dal ramo develop verranno creati i nuovi rami per le nuove funzionalità; se una funzionalità esiste già, è necessario lavorare sul ramo dedicato ad essa;
- è possibile creare un sottoramo per correggere problemi riscontrati in una funzionalità; le correzioni saranno poi integrate al ramo principale della funzionalità.

### Tipi di file e .gitignore

I file utilizzati per la documentazione del progetto sono:

- file con estensione .tex di L<sup>A</sup>T<sub>E</sub>X<sub>G</sub>;
- file con estensione .pdf che sono oggetto di consegna;
- alcuni file testuali e immagini di supporto ai precedenti.

Il file ".gitignore" è presente al livello più esterno del repository<sub>G</sub> ed elenca tutti i file da escludere dal versionamento<sub>G</sub>. È possibile aggiungere il file stesso al suo contenuto, in modo da storicizzare solo i file cui si è interessati.

## Utilizzo di Git<sub>G</sub>

Il repository<sub>G</sub> di Git<sub>G</sub> è composto da vari branch, per lavorare in modo modulare e collaborativo. La procedura<sub>G</sub> consigliata è illustrata di seguito.

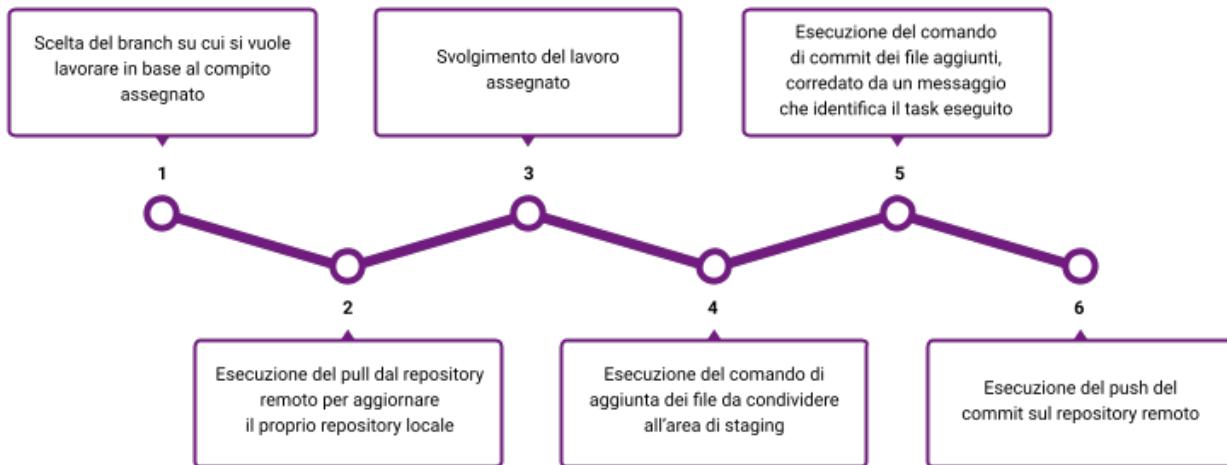


Figura 3: Procedura per l'utilizzo di Git

### 3.2.2.2 Gestione delle modifiche

Tutti i membri del team<sub>G</sub> possono modificare i file in ogni branch, ad eccezione del branch master, per il quale occorre fare una pull request e ottenere l'approvazione di un altro membro. Per effettuare modifiche maggiori sui contenuti o sulla struttura dei file è previsto di:

- contattare il responsabile del file da modificare;
- suggerire la modifica da effettuare;
- se il responsabile valuta positivamente la modifica, allora la applica.

Per modifiche minori, come correzioni grammaticali o miglioramenti sintattici, si consiglia di modificare indipendentemente. In ambo i casi è opportuno commentare i propri commit<sub>G</sub> con chiarezza per risalire facilmente alle modifiche.

### 3.2.3 Metriche

Non sono state attualmente individuate metriche specifiche al processo di gestione della configurazione.

### 3.2.4 Strumenti

Per le parti del progetto da versionare si è scelto di usare il sistema di versionamento<sub>G</sub> distribuito Git<sub>G</sub>, usando il servizio di GitHub<sub>G</sub> per ospitare il repository<sub>G</sub> remoto. Vengono utilizzati i software<sub>G</sub> GitFlow, GitKraken e GitHub<sub>G</sub> desktop per la clonazione del repository in locale.

### 3.3 Gestione della qualità

#### 3.3.1 Descrizione

Lo scopo della gestione della qualità è di garantire che il prodotto e i servizi offerti rispettino gli obiettivi di qualità e che i bisogni del proponente<sub>G</sub> siano soddisfatti. Infatti con la gestione della qualità si intende ottenere:

- qualità nell'organizzazione e nei suoi processi;
- qualità nel prodotto;
- qualità provata oggettivamente;
- soddisfazione finale di cliente e proponente<sub>G</sub>.

Nel *Piano di Qualifica v3.0.0* sono descritte le modalità utilizzate per garantire la qualità nello sviluppo del progetto. In particolare per ogni processo ed ogni prodotto vengono descritti gli obiettivi e le metriche per la valutazione del raggiungimento degli stessi.

#### 3.3.2 Attività

##### 3.3.2.1 Pianificazione

Tutte le attività periodiche di controllo della qualità vengono applicate nel *Piano di Qualifica v3.0.0*. Durante tutta la durata del progetto si fa riferimento a questo documento, il quale viene costantemente aggiornato con nuove metriche e vengono rese disponibili le verifiche realizzate.

##### 3.3.2.2 Garanzia di qualità del prodotto

La qualità del prodotto viene garantita attraverso i processi descritti in §3.5 (Verifica) e §3.6 (Validazione). Questi processi permettono di verificare il rispetto delle metriche di qualità stabilite. Un altro fattore che contribuisce alla qualità del prodotto è il confronto con il proponente<sub>G</sub>, con il quale si può avere un feedback diretto, monitorando la soddisfazione dei requisiti<sub>G</sub> concordati.

##### 3.3.2.3 Garanzia di qualità dei processi

La qualità dei processi viene garantita attraverso lo svolgimento corretto e normato delle attività che compongono tali processi. Viene monitorato lo svolgimento delle attività e il perseguitamento dei principi di efficacia ed efficienza del prodotto durante tutto il suo ciclo di vita. I risultati sono quindi ottenuti rispettando le norme e gli standard che sono stati scelti come riferimento.

##### 3.3.2.4 Classificazione

###### Classificazione delle caratteristiche del prodotto

Le caratteristiche qualitative del prodotto vengono identificate nel modo seguente:

**QC-[Codice]**

Dove:

- **QC**: è un'abbreviazione per *Quality Characteristic*;
- **Codice**: è un numero naturale che identifica la caratteristica di prodotto.

## Classificazione dei processi

Le caratteristiche qualitative di un processo vengono identificate nel modo seguente:

**QP-[Codice]**

Dove:

- **QP**: è un'abbreviazione per *Quality Process*;
- **Codice**: è un numero naturale che identifica il processo.

## Classificazione delle metriche

Le metriche vengono identificate nel modo seguente:

**QM-[Tipo]-[Codice]**

Dove:

- **QM**: è un'abbreviazione per *Quality Metric*.
- **Tipo**: riguarda la tipologia della metrica e può assumere i seguenti valori:
  - **PROC**: se la metrica è riferita a un processo;
  - **PROD**: se la metrica è riferita a una caratteristica di prodotto.
- **Codice**: è un numero naturale che identifica la metrica.

### 3.3.3 Metriche

#### 3.3.3.1 QP-2 Gestione della qualità

Tabella 8: Metriche di processo relative al processo di qualità

Codice	Nome	Descrizione	Formula
QM-PROC-3	Metriche soddisfatte (MS)	È un valore in percentuale che mira a rappresentare la qualità del processo o prodotto analizzato.	$\frac{\# \text{metriche soddisfatte}}{\# \text{totale di metriche}} \times 100$

### 3.3.4 Strumenti

Di seguito sono descritti gli strumenti identificati per il processo di gestione della qualità.

- parte dei processi forniti dallo standard ISO/IEC 12207;
- metriche di qualità stabilite;

## 3.4 Gestione dei cambiamenti

### 3.4.1 Descrizione

Il processo di gestione dei cambiamenti definisce una procedura<sub>G</sub> di analisi e gestione dei problemi. L'obiettivo del processo è fornire una risposta efficace alla rilevazione di un problema, garantendo inoltre la tracciabilità delle modifiche effettuate e dell'insorgenza del problema stesso. Attraverso questo processo si intende inoltre tenere traccia degli errori più comuni, al fine di identificarne le fonti e mettere in atto meccanismi di prevenzione e di miglioramento.

### 3.4.2 Attività

#### 3.4.2.1 Rilevazione del problema

In seguito alla rilevazione di un problema significativo o alla ricezione di una segnalazione dovrà sempre essere aperta una issue<sub>G</sub> all'interno dell'issue tracking system<sub>G</sub>. Una issue è caratterizzata da un codice che identifica, classifica e assegna la priorità alla issue<sub>G</sub> così strutturato:

C-[tipo]-[priorità]-[numero]

dove:

- [tipo] assume valore **P** se la issue riguarda il *prodotto* o **A** se riguarda un *processo*;
- [priorità] assume valore **H**, **M** o **L** a seconda del fatto che la priorità con cui debba essere risolta sia alta (*High*), media (*Medium*) o bassa (*Low*);
- [numero] è un numero intero progressivo e parte da 1.

Ogni issue<sub>G</sub> deve inoltre essere fornita di una descrizione completa ed esaustiva. La descrizione deve permettere a ogni membro del gruppo di capire la natura del problema insorto. Nel caso in cui si ritenga che il problema rilevato abbia priorità elevata e necessiti di una rapida gestione, bisogna avvisare gli altri componenti del gruppo o le parti interessate tramite i canali di comunicazione interna. Un componente del gruppo può prendere in carico una issue<sub>G</sub>, coerentemente con il proprio ruolo e le proprie competenze, notificandolo agli altri componenti tramite l'issue tracking system<sub>G</sub>. Le issue<sub>G</sub> aperte verranno visionate durante gli incontri interni, al fine di monitorare l'andamento del progetto.

#### 3.4.2.2 Risoluzione del problema

I componenti del gruppo che prenderanno in carico la gestione del problema dovranno sfruttare il sistema di collegamento tra issue<sub>G</sub> e commit<sub>G</sub> integrato nell'issue tracking system<sub>G</sub>, al fine di tracciare le modifiche effettuate in risposta ad una segnalazione. In seguito alla risoluzione del problema si completerà la gestione dei cambiamenti tramite la chiusura dell'issue<sub>G</sub> e la storicitizzazione dei cambiamenti adottati per ogni issue insorta.

### 3.4.3 Metriche

Non sono state attualmente individuate metriche specifiche al processo di gestione dei cambiamenti.

### 3.4.4 Strumenti

Per il processo di gestione dei cambiamenti si è deciso di sfruttare l'issue tracking system<sub>G</sub> integrato in GitHub<sub>G</sub> in quanto facilita il collegamento tra issue<sub>G</sub> e commit<sub>G</sub>, consentendo inoltre una facile identificazione dei componenti del team<sub>G</sub> coinvolti.

## 3.5 Verifica

### 3.5.1 Descrizione

L'obiettivo del processo di verifica è controllare la qualità delle attività svolte durante le diverse fasi dello sviluppo, per assicurare che il prodotto derivante da ogni attività sia completo e conforme alle aspettative. In questa fase si cerca di identificare eventuali difetti<sub>G</sub> nel software<sub>G</sub> e nei documenti per poterli correggere, assicurando che il prodotto sia realizzato correttamente. Per realizzare in modo corretto il processo di verifica il team<sub>G</sub> si impegna a:

- eseguire la verifica secondo procedure<sub>G</sub> definite;
- applicare la fase di verifica al termine di ogni attività, non solo a prodotto ultimato;
- seguire criteri tipici e sicuri della fase di verifica;
- eseguire un rapporto finale sui difetti<sub>G</sub> trovati, che verrà aggiunto alla documentazione.

### 3.5.2 Attività

#### 3.5.2.1 Analisi

##### Analisi statica

L'analisi statica è basata sul controllo informale di documenti e codice, senza l'esecuzione del programma, per valutare l'adesione alle norme stabilite e la correttezza del contenuto. L'attività di analisi verrà svolta dal gruppo di revisione<sub>G</sub>, che è diverso dal gruppo di codifica<sub>G</sub> o gruppo di redazione nel caso dei documenti. Il controllo statico è tendenzialmente identificato come un'attività manuale, ma molti controlli statici possono essere eseguiti da analizzatori<sub>G</sub> automatici molto efficienti e rigorosi. Le tecniche più comuni per effettuare l'analisi statica sono il walkthrough<sub>G</sub> e l'inspection<sub>G</sub>. Queste due tecniche si differenziano per il tipo di errori che si prefissano di scoprire. Alcuni difetti<sub>G</sub> comuni nei documenti sono:

- il formato delle date, per le quali è necessario attenersi allo standard scelto;
- il formato degli elenchi puntati;
- i tempi verbali, è preferibile usare il presente;
- formato del documento non rispettato.

##### Analisi dinamica

L'analisi dinamica richiede l'esecuzione del programma in un ambiente di sviluppo<sub>G</sub> e con dati di ingresso controllati, verificando che i risultati prodotti dall'esecuzione del codice siano conformi a quelli in ingresso.

### 3.5.2.2 Test

Sono la costituente principale dell'analisi dinamica e hanno lo scopo di verificare il corretto funzionamento del codice. I test devono essere pensati per poter catturare il maggior numero di difetti<sub>G</sub> possibili, bilanciando il costo con i benefici. Quali e quante prove effettuare è stabilito dal *Piano di Qualifica*. I test devono:

- specificare input e output attesi;
- specificare l'ambiente di sviluppo<sub>G</sub> di esecuzione;
- essere veloci da eseguire;
- essere indipendenti tra loro;
- essere ripetibili;
- essere indipendenti dall'ordine di esecuzione;
- essere automatici, ogni modifica del codice sorgente dovrebbe scatenare l'esecuzione dei test.

#### Classificazione

La classificazione dei test avviene attraverso la seguente forma:

**T[Categoria][Tipologia]-[Importanza]-[Codice]**

Dove:

- **Categoria:** indica la categoria a cui appartiene il test tra quelle descritte nel seguito del paragrafo. Può assumere i seguenti valori:
  - U: test di unità<sub>G</sub>;
  - I: test di integrazione;
  - S: test di sistema;
  - A: test di accettazione.
- **Tipologia:** rappresenta la classe a cui appartiene il requisito<sub>G</sub> sotto test. Può assumere i valori:
  - F: funzionale;
  - P: prestazionale;
  - Q: qualitativo;
  - V: vincolo.
- **Importanza:** indica il grado di importanza del requisito<sub>G</sub> preso in esame. Può assumere i seguenti valori:
  - O: requisito<sub>G</sub> obbligatorio;
  - D: requisito<sub>G</sub> desiderabile;
  - F: requisito<sub>G</sub> facoltativo.
- **Codice:** identificativo numerico del requisito<sub>G</sub> preso in esame.

Nel *Piano di Qualifica v3.0.0* sono elencati in forma tabellare tutti i test pianificati per il progetto. Per ogni test viene riportato il codice identificativo, la descrizione e lo stato, dove lo stato può essere:

- **S:** il test è stato svolto;
- **I:** il test è stato implementato;
- **NI:** il test non è ancora stato implementato.

I test del software<sub>G</sub> sono molteplici e si differenziano per oggetto in esame e scopo.

### Test di unità<sub>G</sub>

Sono i test che verificano le singole unità<sub>G</sub>. Il controllo su un modulo<sub>G</sub> ha come obiettivo principale la correttezza funzionale delle operazioni esportate dal modulo<sub>G</sub>. Fare test sui moduli<sub>G</sub> significa dover effettuare controlli su sistemi che sono incompleti. È necessario che l'ambiente di test preveda dei componenti fintizi che simulino le parti mancanti del sistema. Bisogna testare tutti i possibili input e verificare che vengano prodotti i corretti output. Le singole unità<sub>G</sub> si possono testare con driver<sub>G</sub> e stub<sub>G</sub>. È responsabilità del singolo programmatore scrivere i test delle unità<sub>G</sub> da lui sviluppate. Nel caso di unità<sub>G</sub> complesse il test di unità<sub>G</sub> deve essere eseguito da un verificatore.

### Test di integrazione

Gli obiettivi dei test di integrazione sono la minimizzazione del lavoro e delle risorse necessarie all'integrazione e la massimizzazione del numero di difetti<sub>G</sub> scoperti prima dei controlli sul sistema completo. Le unità<sub>G</sub> che passano il test vengono integrate formando una nuova unità<sub>G</sub>, si procede per passi incrementali fino ad arrivare al sistema completo.

### Test di sistema

Il test di sistema valuta ogni caratteristica di qualità del prodotto software<sub>G</sub> nella sua completezza, avendo come riscontro i requisiti<sub>G</sub> dell'utente. In questa fase ci si assicura che siano rispettate tutte le specifiche definite nell'*Analisi dei Requisiti*. L'obiettivo è verificare che l'interazione tra tutte le componenti del sistema produca i risultati attesi.

### Test di regressione

Il test di regressione si applica in seguito ad una modifica del sistema, per controllare che la sua funzionalità non sia stata compromessa. Prevede la ri-esecuzione dei test esistenti.

### Test di accettazione

Descritto nel paragrafo §3.6.2.2 (Validazione - Test di accettazione).

#### 3.5.3 Metriche

##### 3.5.3.1 QP-3 Verifica

Tabella 9: Metriche di processo relative al processo di verifica

Codice	Nome	Descrizione	Formula
QM-PROC-4	Code coverage (CC)	Percentuale di copertura del codice sorgente da parte dei test.	$\frac{\text{linee percorse dai test}}{\text{linee di codice totali}} \times 100$
QM-PROC-5	Branch coverage (BC)	Percentuale di copertura da parte dei test delle possibili deviazioni di flusso del codice.	$\frac{\text{deviazioni di flusso coperte}}{\text{deviazioni di flusso totali}} \times 100$
QM-PROC-6	Test superati (TS)	Percentuale di test effettuati sul codice superati.	$\frac{\text{test superati}}{\text{test effettuati}} \times 100$

### 3.5.4 Strumenti

Di seguito sono descritti gli strumenti identificati per il processo di verifica.

- **Verifica ortografica:** viene utilizzato l'editor TeXstudio<sub>G</sub>, che segnala in tempo reale gli errori ortografici e le ripetizioni di termini a breve distanza. Ciò garantisce una documentazione sempre corretta. Inoltre tutti i membri del team<sub>G</sub> sono tenuti a conoscere i documenti e prenderli sempre come punto di riferimento, pertanto è concesso a tutti, una volta letti i documenti, suggerire modifiche per quanto riguarda la struttura delle frasi o i termini da utilizzare;
- **Validazione<sub>G</sub> codice:** viene utilizzato il validatore<sub>G</sub> ufficiale fornito da W3C<sub>G</sub> per la validazione<sub>G</sub> del codice scritto in HTML<sub>G</sub> e CSS<sub>G</sub>;
  - Validatore<sub>G</sub> HTML<sub>G</sub>: <http://validator.w3.org/>;
  - Validatore<sub>G</sub> CSS<sub>G</sub>: <http://jigsaw.w3.org/css-validator/>.
- **ESLint<sub>G</sub>:** utilizzato per trovare e risolvere errori presenti nel codice JavaScript<sub>G</sub>. Permette di definire degli standard da seguire e guida lo sviluppatore nella scrittura del codice, avvisandolo di eventuali deviazioni dagli standard stabiliti.

## 3.6 Validazione<sub>G</sub>

### 3.6.1 Descrizione

L'obiettivo del processo di validazione<sub>G</sub> è il controllo di qualità del prodotto rispetto ai requisiti<sub>G</sub> del committente<sub>G</sub> per garantirne la soddisfazione. La validazione<sub>G</sub> è un'attività normalmente prevista sul prodotto finito ma è comunque possibile effettuare delle operazioni di validazione<sub>G</sub> anche durante il processo di sviluppo.

### 3.6.2 Attività

#### 3.6.2.1 Pianificazione

Per gestire il processo di validazione<sub>G</sub> è necessario effettuare una pianificazione, la quale deve focalizzarsi sui seguenti punti essenziali:

- identificare gli oggetti da validare;
- quali operazioni eseguire;
- risorse, responsabilità e gestione delle scadenze legate alla pianificazione.

#### 3.6.2.2 Test

##### Test di accettazione

Tramite questi test si vuole stabilire se il prodotto finito possa essere accettato come soluzione alle reali necessità dell'utente finale. Viene definito anche test di collaudo, perché viene eseguito in un sistema identico o quanto più simile a quello reale, in collaborazione con il committente<sub>G</sub>. Vengono testate le funzionalità principali per attestare il soddisfacimento dei requisiti<sub>G</sub> richiesti. L'obiettivo non è più andare alla ricerca di bugs<sub>G</sub> o problemi, ma ottenere l'accettazione da parte del committente<sub>G</sub> per poter rilasciare il software<sub>G</sub>.

#### 3.6.3 Metriche

Non sono state attualmente individuate metriche specifiche al processo di validazione<sub>G</sub>.

#### 3.6.4 Strumenti

Non sono stati attualmente individuati strumenti specifici al processo di validazione<sub>G</sub>.

## 4 Processi organizzativi

### 4.1 Gestione organizzativa

#### 4.1.1 Descrizione

Il processo di gestione organizzativa descrive le scelte riguardanti la suddivisione e il coordinamento del lavoro all'interno del progetto. Il fine principale di questo processo è fornire ai membri del gruppo un *Piano di Progetto* che permetta di organizzare il lavoro con efficacia ed efficienza. Si tratta quindi di:

- identificare i rischi che si possono verificare durante lo sviluppo del progetto e definirne la gestione, creando un modello organizzativo;
- definire un modello di sviluppo da seguire;
- pianificare le attività da svolgere in base alle scadenze temporali;
- calcolare un preventivo in termini di ore e costi suddiviso per ruoli;
- effettuare un bilancio finale delle spese.

Queste attività sono raccolte nel *Piano di Progetto v3.0.0*, la cui redazione è in carico al responsabile di progetto con la collaborazione dell'amministratore.

#### 4.1.2 Attività

##### 4.1.2.1 Gestione dei ruoli di progetto

Ogni membro del gruppo dovrà ricoprire a rotazione un ruolo corrispondente ad una figura aziendale. Ad ogni ruolo corrisponde un determinato profilo professionale che richiede competenza ed esperienza.

###### Responsabile

Il responsabile ha la responsabilità della pianificazione, del controllo, della gestione e del coordinamento di attività e risorse del progetto. Ricopre inoltre un ruolo di intermediario con il committente<sub>G</sub> e il proponente<sub>G</sub> del capitolato<sub>G</sub>. Si occupa principalmente di:

- pianificare ed emanare scadenze;
- approvare i documenti;
- coordinare le risorse umane;
- gestire e controllare le attività del gruppo;
- gestire ed analizzare le criticità;
- approvare l'offerta sottoposta al committente<sub>G</sub>.

## Amministratore

L'amministratore ha il compito di gestione, supporto e controllo dell'ambiente di lavoro. Si occupa principalmente di:

- controllare e gestire l'efficienza e l'operatività dell'ambiente di sviluppo<sub>G</sub>;
- dirigere le infrastrutture di supporto;
- gestire la documentazione;
- redigere le *Norme di Progetto*;
- collaborare alla redazione del *Piano di Progetto*;
- gestire la configurazione del prodotto;
- redigere ed attuare i piani e le procedure<sub>G</sub> per la gestione della qualità.

## Analista

L'analista è il responsabile delle attività di analisi dei problemi e del dominio applicativo. Egli non è presente all'interno del gruppo per tutta la durata del progetto in quanto si occupa solo delle attività svolte durante l'*Analisi dei Requisiti*. Si occupa principalmente di:

- studio del dominio applicativo del progetto;
- definire la complessità del problema;
- definire i requisiti<sub>G</sub>;
- redigere l'*Analisi dei Requisiti* e lo *Studio di Fattibilità*.

## Progettista

Il progettista si occupa delle attività svolte durante la progettazione dell'architettura e la progettazione di dettaglio. Ha quindi competenze tecniche e tecnologiche. Si occupa principalmente di:

- prendere decisioni efficienti ed efficaci sugli aspetti tecnici del progetto;
- sviluppare l'architettura del prodotto in modo che sia efficiente, efficace e mantenibile. Utilizza apposite tecnologie note ed ottimizzate, individuate a partire dai requisiti<sub>G</sub> definiti dall'analista;
- redigere la specifica tecnica, la definizione di prodotto e la parte pragmatica del *Piano di Qualifica*.

## Programmatore

Il programmatore è responsabile della codifica<sub>G</sub> del progetto e delle componenti che serviranno per effettuare le prove di verifica e validazione. Si occupa principalmente di:

- implementare la specifica e le decisioni del progettista;
- svolgere la parte di codifica<sub>G</sub> del prodotto;
- creare le componenti di supporto per la verifica e validazione del codice.

## Verificatore

Il verificatore è il responsabile dell'attività di verifica dei prodotti di tutte le attività svolte dagli altri membri. È responsabile del controllo sia della documentazione che del codice. Ha il compito di segnalare eventuali errori affidandosi agli standard definiti nelle *Norme di Progetto* e alla propria capacità di giudizio. Si occupa principalmente di:

- esaminare i prodotti in fase di revisione;
- evidenziare eventuali errori;
- segnalare eventuali errori all'autore dell'oggetto in fase di verifica.

### 4.1.2.2 Gestione delle comunicazioni

Le comunicazioni possono svolgersi internamente ed esternamente. Le comunicazioni interne coinvolgono esclusivamente i membri del gruppo mentre quelle esterne coinvolgono anche proponente<sub>G</sub> e committente<sub>G</sub>.

#### Comunicazioni interne

Le comunicazioni interne sono gestite principalmente attraverso tre canali:

- **Telegram<sub>G</sub>**: utilizzando un gruppo privato nel quale è possibile scambiare rapidamente comunicazioni brevi ed informali coinvolgendo l'intero team<sub>G</sub>.
- **Zoom<sub>G</sub>**: utilizzato per lo scambio di comunicazioni più consistenti e tecniche, come aggiornamenti sulle attività o discussioni legate a specifiche attività.
- **GitHub<sub>G</sub>**: utilizzando l'Issue Tracking System<sub>G</sub> di GitHub<sub>G</sub> che consente di aggiornare rapidamente l'intero gruppo sull'avanzamento dei lavori.

#### Comunicazioni esterne

Le comunicazioni esterne sono gestite principalmente dal responsabile di progetto e avvengono attraverso Gmail<sub>G</sub>. Viene utilizzata la casella di posta elettronica creata appositamente per il gruppo *QuaranTeam*, con indirizzo [quaranteam2021@gmail.com](mailto:quaranteam2021@gmail.com) per comunicare con il committente<sub>G</sub> e il proponente<sub>G</sub>.

### 4.1.2.3 Gestione degli incontri

Gli incontri sono indetti dal responsabile, il quale ha il compito di:

- definire la data e l'orario degli incontri considerando la disponibilità dei partecipanti;
- stabilire gli argomenti che verranno trattati durante l'incontro;
- valutare le richieste relative agli incontri da parte dei componenti del gruppo e di eventuali soggetti esterni;
- verificare ed approvare il verbale redatto dal segretario della riunione, il quale viene nominato ad inizio incontro.

I partecipanti sono tenuti ad essere puntuali agli incontri e, nel caso di imprevisti, comunicarli al responsabile. Le decisioni che vengono discusse durante gli incontri sono ritenute approvate nel caso di maggioranza da parte dei partecipanti. Al termine di ciascun incontro viene stilato un verbale da parte del segretario che si è preso carico di appuntare le decisioni prese, le informazioni che sono state scambiate e i programmi per le prossime attività. Tutti gli incontri che non prevedono decisioni non vengono verbalizzati. Vista e considerata la situazione relativa all'emergenza COVID-19<sub>G</sub> gli incontri, sia interni che esterni, si svolgeranno esclusivamente in modalità virtuale tramite video-conferenza.

### Incontri interni

Gli incontri interni sono aperti esclusivamente ai componenti del gruppo *QuaranTeam*. Affinché gli incontri siano ritenuti validi dovranno essere presenti almeno 5 componenti del gruppo. Gli incontri interni si svolgono in video-conferenza sulla piattaforma Zoom<sub>G</sub>.

### Incontri esterni

Gli incontri esterni coinvolgono anche altri soggetti esterni, oltre ai membri del gruppo *QuaranTeam*. Data la presenza di persone esterne, questi incontri assumono una criticità maggiore, che deve essere opportunamente gestita dal responsabile tramite un'adeguata gestione delle comunicazioni che punti a mediare tra il gruppo e gli individui esterni. Ritardi e assenze da parte dei componenti del gruppo sono da evitare. Gli incontri esterni, come quelli interni, avvengono in modalità virtuale e lo strumento di comunicazione utilizzato è Skype<sub>G</sub>. Questo strumento è stato richiesto dal proponente<sub>G</sub> che si è reso disponibile a ricevere su appuntamento tutti i membri del gruppo, o in alternativa tutti i gruppi che concorrono per lo stesso capitolato<sub>G</sub> d'appalto.

#### 4.1.2.4 Gestione degli strumenti di coordinamento

È importante che tutti i componenti del gruppo sappiano, durante lo sviluppo del progetto, quali siano le attività da svolgere, quelle in corso e quelle già completate. Ognuno deve essere a conoscenza dei compiti a lui assegnati e della relativa data di scadenza per il loro completamento. Tutto ciò è utile per poter gestire in maniera efficace ed efficiente il proprio lavoro. Il responsabile deve poter assegnare nuovi compiti ai membri del gruppo e controllarne lo stato di avanzamento per verificarne la coerenza con la pianificazione. Per soddisfare queste necessità si è scelto di utilizzare l'Issue Tracking System<sub>G</sub> fornito da GitHub<sub>G</sub> per il repository<sub>G</sub> usato dal gruppo per il versionamento<sub>G</sub> del progetto. Questo strumento permette di creare delle attività assegnabili, ad una o più persone, attraverso le seguenti informazioni:

- **titolo:** nome del compito da eseguire;
- **descrizione:** descrizione dettagliata del compito da eseguire;
- **assegnatari:** persone a cui compete lo svolgimento del compito;
- **bacheca:** spazio virtuale in cui il compito sarà monitorato;
- **scadenza:** data ultima per lo svolgimento del compito.

Ogni attività attraversa degli stati che permettono di monitorarne l'avanzamento.

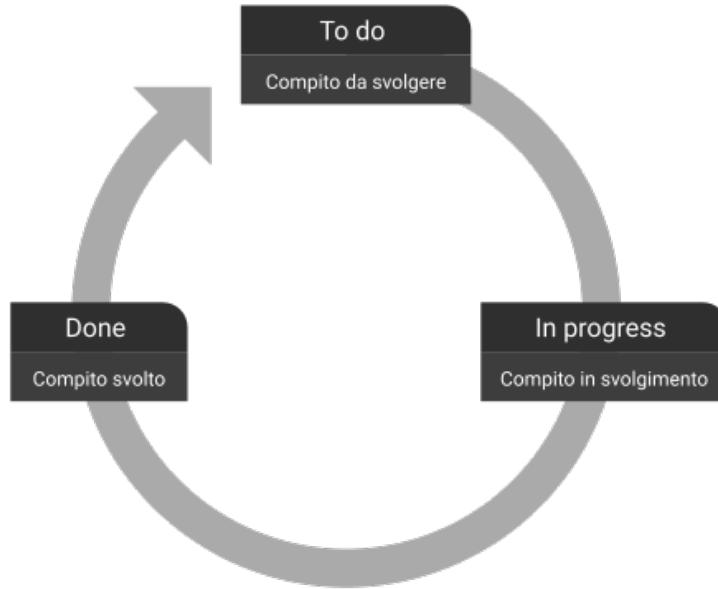


Figura 4: Stati di avanzamento di un'attività nell'Issue Tracking System

La scelta dell'Issue Tracking System<sub>G</sub> di GitHub<sub>G</sub> è data dalla possibilità di gestire il progetto in maniera più semplice e organizzata.

#### 4.1.2.5 Gestione dei rischi

Per la gestione dei rischi viene seguita la seguente procedura<sub>G</sub>.

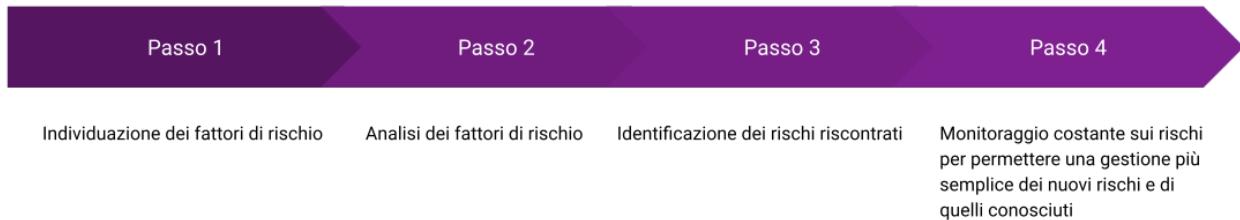


Figura 5: Procedura adottata per la gestione dei rischi

Questa procedura<sub>G</sub> viene applicata nel *Piano di Progetto v3.0.0*, dove viene riportata l'analisi e l'attualizzazione dei rischi. I rischi sono suddivisi principalmente in:

- **Rischi tecnologici:** riguardano i rischi legati allo studio, all'utilizzo e alle funzionalità delle tecnologie software<sub>G</sub> o hardware impiegate durante lo sviluppo del progetto;
- **Rischi organizzativi:** riguardano i rischi legati all'organizzazione generale dello sviluppo e alla gestione del progetto. Sono inclusi i rischi legati al coordinamento dei membri del gruppo e alla gestione dei costi, delle risorse e delle scadenze temporali;
- **Rischi interpersonali:** riguardano i rischi legati ai rapporti tra i membri del gruppo e alla comunicazione interna ed esterna;

- **Rischi legati ai requisiti:** riguardano i rischi dovuti a un'analisi dei requisiti parzialmente scorretta, incompleta o ambigua.

La classificazione dei rischi prevede le seguenti caratteristiche:

- **Occorrenza:** indica una stima approssimativa dell'occorrenza di un rischio e può essere:
  - Alta;
  - Media;
  - Bassa.
- **Gravità:** indica il grado di pericolosità di un rischio e può essere:
  - Alta;
  - Media;
  - Bassa.

#### 4.1.3 Metriche

##### 4.1.3.1 QP-4 Gestione organizzativa

Tabella 10: Metriche di processo relative al processo di gestione organizzativa

Codice	Nome	Descrizione	Formula
QM-PROC-7	Budget at completion (BAC)	È un numero intero che rappresenta il budget inizialmente allocato per la realizzazione del progetto.	
QM-PROC-8	Estimated at completion (EAC)	È un numero intero che rappresenta il budget stimato per la realizzazione del progetto, aggiornato allo stato attuale.	$AC + ETC$
QM-PROC-9	Estimate to complete (ETC)	È un numero intero che rappresenta il budget stimato per la realizzazione delle rimanenti attività necessarie al completamento del progetto.	
QM-PROC-10	Actual cost (AC)	È un numero intero che rappresenta il denaro speso fino al momento del calcolo.	
<i>La tabella continua a pagina seguente.</i>			

Tabella 10: Metriche di processo relative al processo di gestione organizzativa (segue da pagina precedente)

Codice	Nome	Descrizione	Formula
QM-PROC-11	Earned value (EV)	È un numero intero che rappresenta il valore totale del lavoro fatto fino al momento del calcolo.	$\%lavoro\ completato \times BAC$
QM-PROC-12	Planned value (PV)	È un numero intero che rappresenta il denaro che si dovrebbe aver guadagnato in quel momento.	$\%lavoro\ pianificato \times BAC$
QM-PROC-13	Schedule variance (SV)	È un numero intero che indica lo stato dello svolgimento del progetto rispetto alla pianificazione.	$EV - PV$
QM-PROC-14	Cost variance (CV)	È un numero intero che indica il livello di efficienza nello sviluppo del progetto, rispetto a quanto pianificato	$EV - AC$
QM-PROC-15	Rischi non preventivati (RNP)	È un numero intero utilizzato per tracciare in modo incrementale i nuovi rischi che non erano precedentemente preventivati e che si presentano durante una fase del progetto.	Si incrementa di 1 unità per ogni rischio non preventivato individuato.

#### 4.1.4 Strumenti

Di seguito sono descritti gli strumenti identificati dal gruppo per il processo organizzativo.

- **Telegram<sub>G</sub>**: Utilizzato per lo scambio di messaggi informali quali avvisi, domande, consulti e confronti. Utilizzato anche per la comunicazione relativa agli incontri;
- **Gmail<sub>G</sub>**: Utilizzato per comunicare ufficialmente con il committente<sub>G</sub> ed il proponente<sub>G</sub>;
- **Google Drive<sub>G</sub>**: Utilizzato per caricare e visualizzare documenti disponibili a tutti i membri del gruppo in ogni momento e per la stesura di materiale soggetto a molti cambiamenti;
- **Zoom<sub>G</sub>**: Utilizzato per gli incontri a distanza tra i membri del gruppo e con il committente<sub>G</sub>;

- **Skype<sub>G</sub>**: Utilizzato per gli incontri a distanza con il proponente<sub>G</sub>;
- **GitHub<sub>G</sub>**: Utilizzato per il versionamento<sub>G</sub> e il salvataggio in remoto dei file riguardanti il progetto. Sfruttato inoltre l'Issue Tracking System<sub>G</sub> integrato, utilizzato per la gestione e il coordinamento dei compiti da svolgere;
- **Discord<sub>G</sub>**: Utilizzato per lo scambio di messaggi tra i membri del gruppo. Facilita la comunicazione tra sottogruppi interni al team<sub>G</sub> sfruttando la possibilità di creare vari canali dedicati a temi specifici. Consente un'organizzazione più ordinata dei messaggi ed una comunicazione più veloce ed efficace.

## 4.2 Formazione

### 4.2.1 Descrizione

Il processo di formazione ha lo scopo di formare ogni membro del gruppo in modo che abbia le conoscenze adatte per svolgere le attività e utilizzare le tecnologie richieste. La formazione avviene in forma individuale ma il proponente<sub>G</sub> si rende disponibile per eventuali chiarimenti. Ogni membro del gruppo provvede allo studio delle tecnologie utilizzate nel progetto. Ogni persona si documenta sui linguaggi e sugli strumenti di programmazione che vengono utilizzati, operando secondo il principio del miglioramento continuo.

### 4.2.2 Attività

#### 4.2.2.1 Materiale per la formazione

Ogni membro del gruppo è autorizzato a consultare tutto il materiale che risulta necessario all'apprendimento. Viene inoltre utilizzato un canale Discord<sub>G</sub> per la condivisione di materiali e link utili, formando una base di riferimenti e conoscenze comuni che permettano una migliore collaborazione.

Tabella 11: Tecnologie e linguaggi di programmazione utilizzati

Nome	Descrizione
<a href="#"><b>LaTeX</b></a>	Linguaggio di markup <sub>G</sub> per la preparazione di testi.
<a href="#">JavaScript<sub>G</sub></a>	Linguaggio di scripting orientato agli oggetti e agli eventi.
<a href="#">D3.js<sub>G</sub></a>	Libreria <sub>G</sub> JavaScript <sub>G</sub> per creare visualizzazioni dinamiche ed interattive.
<a href="#">Node.js<sub>G</sub></a>	Runtime JavaScript <sub>G</sub> costruito sul motore JavaScript <sub>G</sub> V8 di Chrome.
<a href="#">Express.js<sub>G</sub></a>	Framework <sub>G</sub> per applicazioni web Node.js <sub>G</sub> .
<a href="#">React<sub>G</sub></a>	Libreria <sub>G</sub> JavaScript <sub>G</sub> per la creazione di interfacce utente interattive.
<a href="#">math.js</a>	Libreria <sub>G</sub> matematica per JavaScript <sub>G</sub> e Node.js <sub>G</sub> .
<a href="#">Mocha<sub>G</sub></a>	Framework <sub>G</sub> di test JavaScript <sub>G</sub> eseguito su Node.js <sub>G</sub> e su browser per lo svolgimento di test asincroni.
<a href="#">SonarQube<sub>G</sub></a>	Piattaforma <sub>G</sub> utilizzata per il controllo della qualità del codice.
<a href="#">ESLint<sub>G</sub></a>	Strumento di analisi statica del codice.

#### 4.2.3 Metriche

Non sono state attualmente individuate metriche specifiche al processo di formazione.

#### 4.2.4 Strumenti

Di seguito sono descritti gli strumenti identificati dal gruppo per il processo di formazione.

- **Discord<sub>G</sub>**: utilizzo di un canale apposito per la condivisione di materiali e link utili, permette di avere una base di riferimenti comuni facilmente raggiungibili grazie ad un sistema di tag.