



OUT OF BOUNDS

NORME DI PROGETTO  
PROGETTO CAPTCHA  
Versione 2.0.0

<b>Responsabile</b>	Michele Cazzaro
<b>Redattori</b>	Alberto Matterazzo Simone Bisortole Valentina Caputo Edoardo Retis
<b>Verificatori</b>	Michele Cazzaro Valentina Caputo Simone Bisortole Alberto Matterazzo
<b>Uso</b>	Interno
<b>Destinatari</b>	Out of Bounds prof. Tullio Vardanega prof. Riccardo Cardin

---

CONTATTI

[sweoutofbounds@gmail.com](mailto:sweoutofbounds@gmail.com)

REPOSITORIES

[orgs/SWE-OutOfBounds/repositories](https://github.com/SWE-OutOfBounds/repositories)

## Registro delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
2.0.0	2023/05/27	Michele Cazzaro	Responsabile	Approvazione e rilascio documento
1.1.0	2023/05/26	Alberto Matterazzo	Verificatore	Verifica generale del documento.
1.0.5	2023/05/20	Edoardo Retis, Alberto Matterazzo	Amministratore, Verificatore	Aggiunta e stesura sottosezioni "Strumenti utilizzati" e verifica.
1.0.4	2023/05/15	Edoardo Retis, Alberto Matterazzo	Amministratore, Verificatore	Modifica §3.3.2 e verifica.
1.0.3	2023/05/05	Valentina Caputo, Simone Bisortole	Amministratore, Verificatore	Stesura §2.2.4.2 e verifica.
1.0.2	2023/05/02	Simone Bisortole, Valentina Caputo	Amministratore, Verificatore	Stesura §2.2.4.1 e verifica.
1.0.1	2023/04/18	Alberto Matterazzo, Michele Cazzaro	Amministratore, Verificatore	Stesura §2.2.3.2 e verifica.
1.0.0	2023/02/14	Valentina Caputo	Responsabile	Approvazione e rilascio.
0.4.0	2023/02/12	Jacopo Angeli	Verificatore	Verifica del documento
0.3.3	2023/02/10	Michele Cazzaro	Amministratore	Stesura sezioni §3.2.2, §3.4, §3.5.
0.3.2	2022/12/23	Valentina Caputo	Amministratore	Ampliamento sezione §3.3.
0.3.1	2022/12/22	Valentina Caputo	Amministratore	Correzione errori individuati sezione §1, §2 e §3.

0.3.0	2022/12/20	Jacopo Angeli	Verificatore	Verifica documento
0.2.1	2022/12/12	Edoardo Retis	Amministratore	Correzioni errori e ampliamento sottosezione §2.2.2.1
0.2.0	2022/12/08	Valentina Caputo	Verificatore	Verifica documento
0.1.5	2022/12/05	Alberto Matterazzo	Amministratore	Stesura sezione §2.1 e le sue sottosezioni
0.1.4	2022/11/27	Jacopo Angeli	Amministratore	Ampliamento sezione "Processi Primari", definita convenzione su formato tabelle
0.1.3	2022/11/27	Jacopo Angeli	Amministratore	Aggiunta convenzione "registro delle modifiche"
0.1.2	2022/11/23	Jacopo Angeli	Amministratore	Popolamento Sezioni "Processi Organizzativi" e "Introduzione"
0.1.1	2022/11/21	Jacopo Angeli	Amministratore	Correzione errori
0.1.0	2022/11/16	Edoardo Retis	Verificatore	Verifica del documento
0.0.6	2022/11/14	Jacopo Angeli	Amministratore	Correzione formato e Popolamento Sezioni "Processi di supporto" e "Processi Organizzativi"
0.0.5	2022/11/10	Michele Cazzaro	Amministratore	Ampliamento voce "Strumentazione"
0.0.4	2022/11/09	Jacopo Angeli	Amministratore	Popolamento Sezione "Processi di supporto"
0.0.3	2022/11/08	Jacopo Angeli	Amministratore	Popolamento Sezione "Processi di supporto"

0.0.2	2022/11/05	Jacopo Angeli	Amministratore	Popolamento Sezione "Processi di supporto"
0.0.1	2022/11/05	Edoardo Retis	Amministratore	Creata struttura documento

**Tabella 1:** Registro delle modifiche

# Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo del documento . . . . .	6
1.2	Scopo del capitolato . . . . .	6
1.3	Glossario . . . . .	6
1.4	Riferimenti . . . . .	6
1.4.1	Normativi . . . . .	6
1.4.2	Informativi . . . . .	6
<b>2</b>	<b>Processi primari</b>	<b>7</b>
2.1	Processo di Fornitura . . . . .	7
2.1.1	Scopo . . . . .	7
2.1.2	Descrizione . . . . .	7
2.1.3	Attività . . . . .	7
2.1.3.1	Valutazione della proposta . . . . .	7
2.1.3.2	Preparazione della risposta . . . . .	8
2.1.3.3	Contrattazione . . . . .	8
2.1.3.4	Pianificazione . . . . .	8
2.1.3.5	Esecuzione e controllo . . . . .	9
2.1.3.6	Revisione e valutazione . . . . .	9
2.1.3.7	Consegna e completamento . . . . .	9
2.1.4	Strumenti utilizzati . . . . .	9
2.2	Processo di sviluppo . . . . .	10
2.2.1	Scopo . . . . .	10
2.2.2	Analisi dei requisiti . . . . .	10
2.2.2.1	Classificazione e descrizione dei requisiti . . . . .	10
2.2.2.1.1	Codice identificativo . . . . .	10
2.2.2.1.2	Casi d'uso . . . . .	12
2.2.2.1.3	Codice identificativo . . . . .	12
2.2.2.1.4	Diagramma UML . . . . .	12
2.2.2.2	Strumenti utilizzati . . . . .	13
2.2.3	Progettazione . . . . .	13
2.2.3.1	<i>Requirements &amp; Technology Baseline</i> . . . . .	14
2.2.3.2	<i>Product Baseline</i> . . . . .	14
2.2.3.3	Strumenti utilizzati . . . . .	14
2.2.4	Codifica . . . . .	15
2.2.4.1	Convenzioni e stile . . . . .	15
2.2.4.2	Strumenti utilizzati . . . . .	17
<b>3</b>	<b>Processi di supporto</b>	<b>19</b>
3.1	Documentazione . . . . .	19
3.1.1	Scopo . . . . .	19
3.1.2	Precompilati . . . . .	19
3.1.3	Stati del documento . . . . .	19
3.1.4	Struttura condivisa . . . . .	19
3.1.5	Convezioni . . . . .	20
3.1.5.1	Nomi dei <i>file</i> . . . . .	20

3.1.5.2	Stili di testo . . . . .	20
3.1.5.3	Registro delle modifiche . . . . .	21
3.1.5.4	Tabelle . . . . .	21
3.1.6	Strumenti utilizzati . . . . .	21
3.2	Gestione delle versioni e della configurazione . . . . .	22
3.2.1	Documentazione . . . . .	22
3.2.1.1	Condivisione del materiale . . . . .	22
3.2.1.2	Versionamento . . . . .	22
3.2.1.3	Metriche del codice di versione . . . . .	23
3.2.1.4	Struttura repository . . . . .	23
3.2.2	Software . . . . .	23
3.2.2.1	Condivisione del materiale . . . . .	23
3.2.2.2	Versionamento . . . . .	23
3.2.2.3	Struttura Repository . . . . .	23
3.2.3	Strumenti utilizzati . . . . .	24
3.3	Gestione della qualità . . . . .	24
3.3.1	Misure di qualità . . . . .	24
3.3.2	Strumenti utilizzati . . . . .	24
3.4	Verifica . . . . .	25
3.4.1	Verifica della documentazione . . . . .	25
3.4.2	Verifica del codice . . . . .	25
3.4.2.1	Test . . . . .	26
3.4.2.1.1	Classificazione dei test . . . . .	26
3.4.3	Strumenti utilizzati . . . . .	26
3.5	Validazione . . . . .	27
3.5.1	Attività . . . . .	27
<b>4</b>	<b>Processi Organizzativi</b>	<b>28</b>
4.1	Gestione dei processi . . . . .	28
4.1.1	Ruoli di progetto . . . . .	28
4.1.1.1	Elenco dei ruoli . . . . .	28
4.1.2	Rotazione dei ruoli . . . . .	29
4.1.3	Comunicazione . . . . .	29
4.1.3.1	Testuale . . . . .	29
4.1.3.2	Verbale . . . . .	30
4.1.4	Gestione dei rischi . . . . .	30
4.2	Gestione dell'infrastruttura . . . . .	31
4.2.1	Strumenti di coordinamento . . . . .	31
4.2.2	Strumenti di versionamento . . . . .	31
4.3	Formazione . . . . .	31
4.3.1	Modalità di formazione . . . . .	31

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di raccogliere, catalogare e tenere traccia di tutte le norme e le procedure che devono essere rispettate da ogni membro del gruppo per lo sviluppo del progetto. Riporta regole e descrive procedure che riguardano tutti gli aspetti dell'organizzazione e tutti gli elementi prodotti, siano essi documenti o *software*. Il documento sarà completo al 100% solo alla consegna del prodotto finale, pertanto verrà mantenuto e rivisto nel tempo.

## 1.2 Scopo del capitolato

La richiesta dell'azienda Zucchetti è un servizio *web CAPTCHA<sub>G</sub>*, ovvero un prodotto che discrimini esseri umani da *bot<sub>G</sub>* artificiali. Il tipo di tecnologia e la modalità di sviluppo non sono soggetti ad alcun vincolo.

## 1.3 Glossario

I termini che possono generare dubbi riguardo al loro significato vengono contrassegnati con una lettera G al pedice, a indicare che il termine si può trovare nel documento *Glossario V2.0*.

## 1.4 Riferimenti

### 1.4.1 Normativi

- [Regolamento del progetto didattico](#) [Ultima consultazione: 2023/05/26],
- [Documentazione Capitolato presentato dall'azienda](#) [Ultima consultazione: 2023/05/26],
- [Standard ISO/IEC 12207](#) [Ultima consultazione: 2023/05/26].

### 1.4.2 Informativi

- [LaTeXimpaziente — Guida su linguaggio L<sup>A</sup>T<sub>E</sub>X](#) a cura di Lorenzo Pantieri [Ultima consultazione: 2023/05/26],
- [Materiale distribuito nel corso "INGEGNERIA DEL SOFTWARE 2022-2023"](#) [Ultima consultazione: 2023/05/26].

## 2 Processi primari

### 2.1 Processo di Fornitura

#### 2.1.1 Scopo

Il processo di fornitura descrive le principali attività e i compiti necessari alla realizzazione del progetto, al fine di comprendere al meglio le richieste del proponente. Lo scopo del processo di fornitura riguarda la gestione del rapporto con il cliente: comprenderne i bisogni per definire i requisiti del sistema da realizzare in modo da soddisfare pienamente le aspettative.

#### 2.1.2 Descrizione

Durante il processo di fornitura viene valutata la fattibilità e i rischi del prodotto desiderato rispetto alle tecnologie da utilizzare, ai tempi a disposizione e alle competenze del gruppo, vengono formalizzate le procedure da seguire e vengono definiti gli *standard* di qualità che il gruppo intende rispettare.

#### 2.1.3 Attività

Le attività caratterizzanti il processo di fornitura sono le seguenti:

- Valutazione della proposta,
- Preparazione della risposta,
- Contrattazione,
- Pianificazione,
- Esecuzione e controllo,
- Revisione e valutazione,
- Consegna e completamento.

##### 2.1.3.1 Valutazione della proposta

Gli analisti scrivono il documento a uso interno "Analisi dei Capitolati" in cui sono sottoposti a giudizio critico e tecnico gli aspetti di ogni capitolato. Per ogni capitolato vengono trattati i seguenti aspetti:

- **Informazioni generali:** nome del progetto e il proponente,
- **Descrizione:** descrizione tecnica del prodotto da realizzare e un elenco delle caratteristiche del prodotto,
- **Aspetti positivi:** aspetti qualitativi interessanti e stimolanti del capitolato,
- **Criticità e fattori di rischio:** problemi riscontrati nel capitolato e nella fattibilità del progetto,



- **Conclusioni:** bilancio quantificato tra aspetti positivi e problematiche ed eventuali considerazioni utili con i motivi per cui il gruppo ha deciso di accettare o scartare il capitolato.

Dato che nel contesto del progetto accademico sono presenti più capitolati, il gruppo si candida allo sviluppo del prodotto il cui studio di fattibilità ha il miglior bilancio conclusivo.

#### 2.1.3.2 Preparazione della risposta

Il gruppo redige per il committente la lettera di candidatura per candidarsi ufficialmente alla realizzazione del progetto scelto.

#### 2.1.3.3 Contrattazione

Con l'aggiudicazione dell'appalto il gruppo *Out of Bounds* stipula un contratto con il committente per fornire e realizzare il prodotto *software* del proponente del capitolato C1 Zucchetti S.p.A..

#### 2.1.3.4 Pianificazione

Il responsabile di progetto redige il documento "Piano di Progetto" che è composto dalle seguenti sezioni:

- **Analisi dei rischi:** vengono analizzati i rischi, quantificati in termini di pericolosità, che potrebbero verificarsi durante lo svolgimento del progetto, la probabilità di occorrenza e le conseguenze nel caso in cui si verifichino. Vengono descritte precauzioni e il piano di contingenza,
- **Modello di sviluppo:** viene descritto il modello di sviluppo adottato e le motivazioni di tale scelte,
- **Pianificazione:** vengono pianificate le varie attività da svolgere nelle diverse fasi del progetto con le relative scadenze temporali,
- **Preventivo:** viene effettuata una stima delle ore di lavoro per ciascuna fase proponendo così un preventivo,
- **Consuntivo:** vengono indicate le ore effettive di lavoro svolto in ogni fase, con i relativi costi, che vengono confrontati con i costi preventivati.

Gli amministratori redigono il documento *Piano di qualifica* nel quale vengono descritte le strategie di verifica e validazione da adottare per garantire la qualità di prodotto e di processo. Il Piano di Qualifica è composto dalle seguenti sezioni:

- Qualità di processo,
- Qualità di prodotto,
- Specifica dei test,
- Resoconto attività di verifica.

### 2.1.3.5 Esecuzione e controllo

Il gruppo si impegna a:

- Mettere in atto la pianificazione sopracitata,
- Sviluppare il prodotto secondo quanto stabilito,
- Perseguire gli obiettivi di qualità stabiliti nel *Piano di Qualifica*,
- Verificare di frequente lo stato di avanzamento del progetto,
- Tenere traccia di eventuali problematiche e strategie per risolverle.

### 2.1.3.6 Revisione e valutazione

Il gruppo intende mantenere con il cliente un rapporto di collaborazione continuo con lo scopo di approfondire le richieste specifiche del prodotto e di verificare che tali richieste vengano soddisfatte nei tempi e nei modi stabiliti.

L'incontro con il proponente avviene ogni volta che il gruppo lo ritiene necessario per sciogliere dubbi o per mostrare un notevole avanzamento del prodotto. Il gruppo durante l'arco del progetto è sottoposto a delle revisioni intermedie di avanzamento con valutazione da parte del committente: la *Requirements & Technology Baseline<sub>G</sub>* e la *Product Baseline<sub>G</sub>*.

### 2.1.3.7 Consegna e completamento

Il gruppo consegna il prodotto al cliente come specificato dal contratto. La *Product Baseline<sub>G</sub>* è la revisione finale in cui il gruppo consegna e presenta il prodotto, alla presenza sia del committente che del proponente.

### 2.1.4 Strumenti utilizzati

Nome	Descrizione	Riferimenti
Fogli <i>Google</i>	Fogli <i>Google</i> è un servizio sviluppato da <i>Google</i> , che permette di creare, modificare e condividere fogli di lavoro. È stato utilizzato per la creazione di grafici e tabelle organizzative.	<a href="http://www.google.it">www.google.it</a>
<i>Jira</i>	<i>Jira Software</i> è pronto all'uso con le funzioni e le <i>best practice</i> di cui i team <i>Agile</i> hanno bisogno per sviluppare e migliorare le loro pratiche <i>Agile</i> . È stato utilizzato per gestione e il tracciamento del lavoro svolto.	<a href="http://www.atlassian.com">www.atlassian.com</a>

<i>Github</i>	<i>Github</i> è un servizio di hosting per progetti software, di proprietà della società <i>GitHub Inc.</i> , è stato utilizzato per gestione e il tracciamento del lavoro svolto.	<a href="https://github.com">https://github.com</a>
---------------	--	---

**Tabella 2:** Principali strumenti utilizzati nel processo di fornitura.

## 2.2 Processo di sviluppo

### 2.2.1 Scopo

Il processo di sviluppo consiste in tutte le attività da svolgere durante lo sviluppo del prodotto *software*. Questa sezione descrive allora tutte le attività, le norme e le convenzioni adottate durante il processo di sviluppo del prodotto richiesto dal richiedente.

Il processo di sviluppo si compone di tre attività principali:

- Analisi dei requisiti,
- Progettazione,
- Codifica.

### 2.2.2 Analisi dei requisiti

Questa attività ha lo scopo d'individuare e descrivere, in seguito a studio approfondito, i requisiti che il progetto richiede, siano essi espliciti o impliciti, e i casi d'uso che il prodotto deve affrontare.

L'individuazione dei requisiti avviene mediante analisi approfondita del capitolato proposto, confronti diretti con proponente e studio delle tecnologie presenti. L'approccio migliore è quello di suddividere il problema in requisiti sempre più elementari, per diminuirne la complessità del problema iniziale, e affiancare a ogni requisito scomposto un caso d'uso scritto con un diagramma UML<sub>G</sub> per studiarne la realizzazione.

#### 2.2.2.1 Classificazione e descrizione dei requisiti

I requisiti vengono raccolti e descritti nel documento *Analisi dei requisiti*, la loro classificazione avviene mediante utilizzo di codici identificativi.

##### 2.2.2.1.1 Codice identificativo

**R[Necessità][Tipologia][Identificativo]**

dove:

- Necessità: indica il grado di necessità del requisito, viene espresso usando una cifra singola compresa tra 1 e 3 dove:

Valore	Significato
1	Obbligatorio
2	Desiderabile
3	Opzionale

**Tabella 3:** Indice di obbligatorietà dei requisiti

- Tipologia: descrive la tipologia del requisito e la esprime utilizzando un carattere letterale in formato maiuscolo che può assumere i seguenti valori:

Valore	Significato
V	Vincolo
P	Prestazionale
Q	Qualitativo
F	Funzionale

**Tabella 4:** Tipologie di requisiti

- Identificativo: codice identificativo, quindi univoco espresso in forma gerarchica padre/figlio.

Ogni requisito viene riportato con informazioni specifiche quali:

- Descrizione: breve esposizione sul requisito,
- Fonte: origine del requisito.

### Esempio

Requisito	Descrizione	Fonte
R1V1	Descrizione	Capitolato

**Tabella 5:** Esempio di classificazione di requisito



**2.2.2.1.2 Casi d'uso** I casi d'uso esprimono un'interazione tra un soggetto attore e un sistema. La loro definizione viene effettuata nel documento *Analisi dei requisiti V1.0*.

Essi sono costituiti da:

- Codice identificativo,
- Diagramma UML<sub>G</sub>,
- Attori,
- Descrizione,
- Precondizioni,
- Post condizioni,
- Scenario principale,
- Estensioni.

### 2.2.2.1.3 Codice identificativo

Il codice identificativo rispetta la seguente sintassi:

**UC[Numero caso d'uso padre](.[Numero caso d'uso figlio])\* - [Nome del caso d'uso]**

dove:

- UC: indica "Use Case",
- Numero caso d'uso padre: numero identificativo del caso d'uso associato al requisito principale, viene rappresentato da una stringa numerica che viene incrementata a ogni caso d'uso identificato,
- Numero caso d'uso figlio: numero identificativo del caso d'uso associato al sotto caso del requisito principale, viene rappresentato da una stringa numerica che viene incrementata a ogni caso d'uso identificato,
- Nome del caso d'uso: breve titolo descrittivo del caso d'uso.

### 2.2.2.1.4 Diagramma UML

Il diagramma UML<sub>G</sub> viene realizzato utilizzando la versione 2.0 del linguaggio.

#### Esempio

UC1 - Registrazione



**Figura 1:** UC1 - Registrazione

- **Attori:** utente non registrato,
- **Descrizione:** l'utente non registrato compila i campi richiesti per registrarsi nel sistema,
- **Precondizione<sub>G</sub>:** il sistema considera l'utilizzatore come un utente non registrato, il quale desidera procedere con la registrazione,
- **Post condizione<sub>G</sub>:** l'utente ha compilato i campi richiesti dalla procedura di registrazione,
- **Scenario principale:**
  1. L'utente inserisce il suo nome (UC1.1),
  2. L'utente inserisce il suo cognome (UC1.2),
  3. L'utente inserisce la sua *email* (UC1.3),
  4. L'utente inserisce il suo *username* (UC1.4),
  5. L'utente inserisce la sua *password* (UC1.5),
  6. L'utente conferma e invia i dati inseriti (UC1.6).

#### 2.2.2.2 Strumenti utilizzati

Nome	Descrizione	Riferimenti
<i>StarUML</i>	<i>Software</i> che permette la creazione e la modifica di diverse tipologie di diagrammi. È stato utilizzato per la creazione e la manutenzione di diagrammi UML dei casi d'uso.	<a href="http://staruml.io">staruml.io</a>

**Tabella 6:** Principali strumenti utilizzati nel processo di fornitura.

#### 2.2.3 Progettazione

L'attività di progettazione utilizza i risultati prodotti durante l'*analisi dei requisiti* per produrre un prodotto che soddisfi i bisogni e le aspettative del proponente. In questa attività viene decisa l'architettura del sistema, inizialmente realizzata da un prodotto *software* prototipale, chiamato *Proof of Concept<sub>G</sub>*, per fornire una solida base alla *Requirements & Technology Baseline<sub>G</sub>* e successivamente approfondita e sviluppata in dettaglio alla *Product Baseline<sub>G</sub>*.

### 2.2.3.1 *Requirements & Technology Baseline*

Questa fase è necessaria per definire in modo concreto sia i requisiti che il proponente presenta, sia le modalità, le tecnologie, i *framework<sub>G</sub>* e le librerie utilizzate dal fornitore per il corretto rispetto dei requisiti individuati. Viene tutto accompagnato da dimostrazioni accurate di adeguatezza e fattibilità.

Durante questa fase vengono creati nello specifico i seguenti documenti:

- Piano di progetto,
- Piano di qualifica,
- Norme di progetto,
- Analisi dei requisiti,
- Glossario,
- Verbali di riunione.

A esclusione dei *Verbali di riunione*, tutti i documenti sopra citati vengono mantenuti e ampliati anche nelle fasi successive.

### 2.2.3.2 *Product Baseline*

Questa fase segue quella di *Requirements & Technology Baseline<sub>G</sub>*. Questa fase attua il design definitivo, coerente con il *PoC<sub>G</sub>*, ma migliorativo rispetto ad esso. Inoltre in questa fase avviene un avanzamento sostanziale di codifica in modo da avere un prodotto dimostrabile. L'obiettivo di questa fase è quello di realizzare l'*MVP<sub>G</sub>*, ossia una versione preliminare del prodotto atteso dotata di funzionalità sufficienti per valutare la bontà della visione iniziale, consentire un adeguato uso esplorativo e permettere futuri avanzamenti.

Durante questa fase vengono ampliati i documenti nominati nella fase di *Requirements & Technology Baseline<sub>G</sub>* e vengono prodotti due nuovi documenti tecnici esterni come la Specifica Tecnica e il Manuale Utente.

### 2.2.3.3 Strumenti utilizzati

Nome	Descrizione	Riferimenti
<i>StarUML</i>	Software che permette la creazione e la modifica di diverse tipologie di diagrammi. È stato utilizzato per la creazione e la manutenzione di diagrammi delle classi.	<a href="http://staruml.io">staruml.io</a>

**Tabella 7:** Principali strumenti utilizzati nel processo di fornitura.

## 2.2.4 Codifica

L'attività di codifica viene svolta dai programmatori e ha lo scopo di concretizzare la progettazione con la programmazione del software vero e proprio. Nello svolgimento di quest'attività si devono seguire delle norme e convenzioni per generare un codice leggibile e uniforme.

Gli obiettivi di quest'attività sono quelli di agevolare le fasi di verifica, validazione e manutenzione, fornire un prodotto che soddisfi le aspettative del proponente e realizzare un prodotto di qualità facilmente mantenibile ed estendibile.

### 2.2.4.1 Convenzioni e stile

Al fine di garantire l'uniformità, leggibilità e manutenibilità del codice, vengono di seguito elencate:

- **Indentazione:** data l'estrema differenza che può presentare la formattazione del codice generato da sviluppatori differenti, sono state stabilite in comune accordo, una lista di estensioni<sub>G</sub> da utilizzare durante la stesura del codice. La formattazione del codice viene quindi delegata a strumenti automatici che ne garantiscono, a patto di scelta ponderata, uniformità e leggibilità del codice. Il dettaglio degli strumenti citati si può ritrovare nella sezione apposita (2.2.4.2),
- **Nomi di metodi, variabili e costanti:**
  - **Utilizzo di "lowerCamelCase"<sub>G</sub>:** nei nomi composti da più parole, a partire dalla seconda, ogni parola deve iniziare con la lettera maiuscola,
  - **Nomi descrittivi e coincisi:** al fine di limitare la necessità di commenti ogni nome deve essere quanto più descrittivo e conciso possibile, ecco degli esempi:

Scopo dell'oggetto	Da evitare	Preferibile
Rappresentare una lista di utenti	const c	const customers
Contenere una lista di temperature	const temp	const temperature
Contenere l'età di alcuni clienti	const c.age; const cAge	const customersAge

**Tabella 8:** Esempi nella dichiarazione di oggetti durante la fase di codifica.

- **Nomi delle classi:** tutte le parole che compongono un nome di classe iniziano con la lettera maiuscola. Esempi: Decorator, DistortionDecorator,
- **Commenti al codice:**
  - **JsDoc:** deve essere, dove possibile, utilizzato il linguaggio [JSDoc](#), che attribuisce al codice un alto grado di leggibilità e manutenibilità,





- **Commenti specifici:** i commenti all'interno del codice devono essere utilizzati come chiarimento del funzionamento del codice, il loro contenuto deve essere chiaro e sintetico. A questo scopo vengono utilizzati strumenti di intelligenza artificiale, descritti nella sezione apposita 2.2.4.2, affiancati a supervisione attiva.

#### 2.2.4.2 Strumenti utilizzati

Nome	Descrizione	Riferimenti
Ambiente di sviluppo ed estensioni utilizzate		
<i>Visual Studio Code</i>	<i>IDE</i> versatile ed estendibile che utilizza estensioni proprie o sviluppate da terzi	<a href="https://code.visualstudio.com">code.visualstudio.com</a>
<i>Prettier</i>	Estensione specifica per la formattazione automatica del codice.	<a href="https://prettier.io">prettier.io</a>
Principali tecnologie utilizzate		
<i>Node.js</i>	<i>Node.js</i> è un ambiente <i>runtime single-threaded</i> , <i>open-source</i> e multiplatforma per la creazione di applicazioni <i>server-side</i> e di <i>networking</i> veloci e scalabili.	<a href="https://nodejs.org">nodejs.org</a>
<i>Express.js</i>	<i>Express</i> è un <i>framework</i> per applicazioni web <i>Node.js</i> flessibile e leggero che fornisce una serie di funzioni avanzate per le applicazioni web e per dispositivi mobili.	<a href="https://expressjs.com">expressjs.com</a>
<i>NPM</i>	<i>NPM</i> è il più grande registro <i>software</i> del mondo, consente di reperire librerie scritte da aziende o utenti in modo molto rapido mediante <i>CLI</i> .	<a href="https://docs.npmjs.com">docs.npmjs.com</a>
<i>Angular15</i>	Il più utilizzato <i>framework</i> per lo sviluppo di applicazioni web.	<a href="https://angular.io">angular.io</a>
<i>MySQL</i>	<i>MySQL</i> è un sistema <i>open source</i> di gestione di database relazionali <i>SQL</i> sviluppato e supportato da <i>Oracle</i> .	<a href="https://www.mysql.com">www.mysql.com</a>
Principali librerie utilizzate		
<i>Angular Material</i>	<i>Angular Material</i> è un'implementazione di <i>Google Material Design in Angular</i> . Si tratta quindi di una libreria di componenti dell'interfaccia utente per gli sviluppatori che usano <i>Angular</i> .	<a href="https://material.angular.io">material.angular.io</a>
<i>node-canvas</i>	Implementazione dell'oggetto <i>canvas</i> , per ambiente <i>Node.js</i> che utilizza il <i>framework cairo</i> .	<a href="https://www.npmjs.com">www.npmjs.com</a>

<i>bcrypt</i>	Libreria che utilizza l'algoritmo omonimo per mascherare una stringa, tipicamente utilizzata nella gestione delle <i>password</i>	<a href="http://www.npmjs.com">www.npmjs.com</a>
<i>JsonWebToken</i>	Libreria utilizzata per firmare una serie di dati utilizzando una <i>password</i> .	<a href="http://www.npmjs.com">www.npmjs.com</a>
<i>Swagger</i>	<i>Swagger</i> è un insieme di strumenti e tecnologie per la progettazione, la costruzione e la documentazione delle <i>API<sub>G</sub> RESTful</i> .	<a href="http://swagger.io">swagger.io</a>
Altri strumenti		
<i>ChatGPT</i>	<i>ChatGPT</i> è un <i>chatbot</i> basato su intelligenza artificiale e apprendimento automatico sviluppato da <a href="https://openai.com">OpenAI</a> specializzato nella conversazione con un utente umano.	<a href="https://chat.openai.com">chat.openai.com</a>

**Tabella 9:** Principali strumenti utilizzati nella fase di codifica del prodotto.

## 3 Processi di supporto

Questa sezione contiene le regole dei processi coinvolti nell'avanzamento di progetto. Quest'ultime servono a mantenere un ambiente di lavoro ordinato, pertanto verranno seguite da tutti i componenti del gruppo e serviranno come criteri di valutazione in fase di verifica.

### 3.1 Documentazione

#### 3.1.1 Scopo

Questa sezione descrive dettagliatamente tutte le norme e le convenzioni relative alla documentazione prodotta dal gruppo *Out of Bounds* durante l'intera durata del progetto con lo scopo di delineare una valida e chiara struttura per tutti i documenti prodotti dal gruppo stesso.

#### 3.1.2 Precompilati

Sono stati creati dei modelli precompilati così da diminuire drasticamente il tempo utilizzato nella produzione di documentazione. L'utilizzo di quest'ultimi consente inoltre di mantenere una strutturazione dei documenti coerente.

#### 3.1.3 Stati del documento

Ogni documento prodotto durante il progetto si trova sempre in una singola tra le seguenti fasi:

- Stesura iniziale: il documento viene creato e strutturato con le sezioni necessarie (alla data della stesura),
- Popolamento e manutenzione dei contenuti: le sezioni vengono compilate,
- Verifica: il contenuto del documento viene verificato a cadenza regolare da personale incaricato che analizza e conferma la validità del suo contenuto.
- Rilascio: il contenuto del documento viene approvato dal responsabile e quindi rilasciato.

Il fatto che gli stati siano mutualmente esclusivi non implica che il documento possa trovarsi più volte nello stesso stato.

#### 3.1.4 Struttura condivisa

Tutti i documenti prodotti durante l'avanzamento del progetto condividono la medesima struttura:

- Prima pagina: contiene in ordine:
  - Logo del gruppo: centrato e collocato in cima alla pagina,
  - Nome del gruppo: centrato e posto al di sotto del logo,
  - Nome del documento: centrato e posto al di sotto dell'elemento precedente ma con una spaziatura di 2cm,
  - Nome del progetto (Progetto *CAPTCHA*): centrato e posizionato subito dopo l'elemento precedente,
  - Versione di rilascio: vedi elemento precedente,

- Tabella informativa sul documento: centrata e posizionata 3cm dopo l'elemento precedente,
- Sezione contatti e *link* a *repository<sub>G</sub>* del gruppo: posizionata come piè di pagina.
- Registro delle modifiche: contiene ordinate in ordine decrescente per data, tutte le modifiche effettuate al documento con relativi dettagli.  
Questa sezione NON è presente nei verbali di riunione in quanto la redazione del verbale è assegnata a turno a un singolo componente del gruppo, così come la verifica.
- Indice: dopo il registro delle modifiche è presente l'indice dei contenuti con lista delle immagini (se presenti) e lista delle tabelle (se presenti ed escluso il Registro delle modifiche),
- Pagine di contenuto:
  - Intestazione: logo e nome del gruppo e nome del documento,
  - Corpo: contenuto del documento,
  - Piè di pagina: riflettore di posta elettronica del gruppo e numerazione della pagina.

### 3.1.5 Convezioni

#### 3.1.5.1 Nomi dei *file*

La nomenclatura dei file utilizza il *PascalCasing<sub>G</sub>*. Solo nel caso di verbali, dopo il nome del file è presente la data in formato "YYYYMMDD" preceduta dal carattere "\_",

#### 3.1.5.2 Stili di testo

Nei documenti vengono usati diversi stili con funzioni specifiche:

- Collegamenti ipertestuali: si utilizza il colore blu,
- Parole non in lingua italiana: viene usato il corsivo,
- Nome di documenti: si usa il corsivo e viene sempre indicato in seguito al nome anche la versione a cui si fa riferimento.
- Rimandi a sezioni interne al documento: indicate usando il sottolineato,
- Rimandi al glossario: a pedice viene indicata "G",
- Elenchi puntati e numerati:
  - Ogni punto dell'elenco inizia con la lettera maiuscola,
  - Ogni punto (escluso l'ultimo) termina con una virgola,
  - L'ultimo punto dell'elenco termina con un punto,
  - Se l'elenco è una lista di definizioni viene scritto il testo della definizione dopo il termine separata da ": ".
- Date: all'interno dei documenti le date vengono indicate in formato "YYYY-MM-DD",
- Immagini: l'immagine viene seguita da una breve descrizione e viene posta centrata relativamente alla pagina. A queste regole sono sottoposti anche eventuali diagrammi.

### 3.1.5.3 Registro delle modifiche

Il registro delle modifiche contiene tutte le variazioni che subisce il documento. Ogni riga della tabella contiene in ordine:

- Versione del documento,
- Data,
- Autore,
- Ruolo del componente del gruppo,
- Descrizione della modifica.

I cambiamenti riportati sono ordinati in ordine temporale decrescente, così che la prima riga contenga sempre l'ultima operazione effettuata al documento.

### 3.1.5.4 Tabelle

Ogni tabella è centrata orizzontalmente.

L'intestazione è di colore rosso e le scritte sono di colore bianco, mentre per tutte le altre righe il colore di sfondo è bianco mentre il testo è nero, tutte le linee sono solide e di colore nero.

Ogni tabella è seguita dal suo numero in riferimento alle altre tabelle del documento e da una breve descrizione del suo contenuto.

Se sono presenti prezzi sono espressi in euro, sono indicati anche i decimi e i centesimi che sono divisi dalla parte intera da ", " e sono preceduti dal simbolo "€".

Giorno	Prezzo
Lunedì	€50,00
Martedì	Gratis

**Tabella 10:** Tabella d'esempio

### 3.1.6 Strumenti utilizzati

Nome	Descrizione	Riferimenti
<i>LanguageTool</i>	<i>Software</i> che evidenzia errori grammaticali e logici effettuati durante la stesura di testo. Utilizzato in fase di stesura e verifica dei documenti.	<a href="https://www.languagetool.org">languagetool.org</a>
<i>Overleaf</i>	<i>Web editor</i> $\LaTeX$ gratuito che permette la creazione e la modifica, anche contemporanea di file testuali utilizzando il linguaggio $\LaTeX$ .	<a href="https://www.overleaf.com">www.overleaf.com</a>

<i>Visual Studio Code<sub>G</sub></i>	<i>Editor Open-Source.</i>	<a href="https://code.visualstudio.com">code.visualstudio.com</a>
<i>GitHub<sub>G</sub></i>	Piattaforma integrata per versionamento che utilizza <i>Git<sub>G</sub></i> e che fornisce strumenti gestionali per <i>team</i> di sviluppo.	<a href="https://github.com">github.com</a>

**Tabella 11:** Principali strumenti utilizzati nel processo di creazione e manutenzione della documentazione.

## 3.2 Gestione delle versioni e della configurazione

La sezione descrive le modalità e gli strumenti utilizzati nella completa gestione della documentazione e dei prodotti software prodotti durante il corso del progetto didattico.

### 3.2.1 Documentazione

#### 3.2.1.1 Condivisione del materiale

I documenti prodotti vengono mantenuti e modificati su *Overleaf*, così come indicato nella sezione Strumentazione (3.1.6).

A cadenza regolare e a ogni *release* i documenti vengono salvati (se prima versione) o aggiornati (se versioni successive) su *repositories<sub>G</sub> Git<sub>G</sub>*, utilizzando il servizio gratuito *GitHub<sub>G</sub>*. La scelta sull'utilizzo di *GitHub<sub>G</sub>* è dovuta a:

- La maggior parte dei componenti del gruppo ha usato il servizio in precedenza, i restanti componenti invece si sono istruiti all'utilizzo in parte autonomamente, in parte aiutati dai colleghi,
- Il servizio fornisce l'utilissima funzione di sistema di tracciamento dei problemi e uno strumento di gestione temporale che il gruppo utilizzerà come strumenti principale.

In particolare si è deciso di creare una cartella privata dove mantenere i file a uso interno al gruppo e una seconda [Pubblica](#) da condividere con proponente.

#### 3.2.1.2 Versionamento

Ogni documento è identificato da un codice numerico di due cifre

$$[X].[Y]$$

- X: numero della versione stabile, soggetta ad approvazione del responsabile.
- Y: numero della modifica effettuata, soggetta a verifica dal verificatore (*Hot Fix<sub>G</sub>*).

Nella tabella "Registro delle modifiche" compare un'ulteriore cifra che indica una versione ancora da verificare del documento, utile solo internamente al team in quanto il documento verrà rilasciato solo in versione stabile e se necessario con correzioni SEMPRE verificate.



### 3.2.1.3 Metriche del codice di versione

La convenzione adottata per il versionamento della documentazione implica una modalità di avanzamento specifica:

- Le cifre iniziano dal valore zero,
- Versione rilasciata:
  - Rilascio: la prima cifra viene avanzata di un'unità, la seconda cifra viene azzerata,
  - Verifica: la seconda cifra viene avanzata di un'unità.
- Registro delle modifiche
  - Rilascio: la prima cifra viene avanzata di un'unità, la seconda cifra viene azzerata così come la terza,
  - Verifica: la seconda cifra viene avanzata di un'unità, la terza portata al valore zero,
  - Modifica: la terza cifra viene avanzata di un'unità.

### 3.2.1.4 Struttura repository

Contiene tutti i documenti a uso esterno al gruppo in formato Pdf e opportunamente diviso in cartelle

[SWE-OutOfBounds/CaptchaProject:](#)

## 3.2.2 Software

### 3.2.2.1 Condivisione del materiale

Ogni prodotto *software* prodotto viene mantenuto su una *Repository<sub>G</sub> Git<sub>G</sub>* dedicata.

### 3.2.2.2 Versionamento

Il servizio *Git<sub>G</sub>* permette di avere diverse linee di produzione di un prodotto attive allo stesso momento chiamate *branch<sub>G</sub>*. Questa caratteristica ci permette di eseguire al meglio il versionamento.

### 3.2.2.3 Struttura Repository

Ogni *Repository<sub>G</sub>* avrà i seguenti *branch<sub>G</sub>* :

- **main**: ultima versione stabile del prodotto corredata di un file *README.md* dove verranno riportate informazione specifiche sulla versione corrente. Dal quale può essere creato uno o più *branch<sub>G</sub>*:
  - **bug**: *bug/*, usato per risolvere uno specifico *bug<sub>G</sub>* riscontrato nella fase di test,
- **dev**: ultima versione prodotta dal team di sviluppo ancora da verificare e testare completamente. Dal quale possono essere creati:
  - **feat**: *feat/*, usato per lavorare all'aggiunta di una funzione specifica,
  - **junk**: *junk/*. usato per sperimentare.



### 3.2.3 Strumenti utilizzati

Nome	Descrizione	Riferimenti
<i>GitHub<sub>G</sub></i>	Piattaforma integrata per versionamento che utilizza <i>Git<sub>G</sub></i> e che fornisce strumenti gestionali per <i>team</i> di sviluppo.	<a href="https://github.com">github.com</a>

**Tabella 12:** Principali strumenti utilizzati nel processo di creazione e manutenzione della documentazione.

## 3.3 Gestione della qualità

Il gruppo si impegna a gestire la qualità delle produzioni, sia esse di tipo documentale, sia esse di tipo *software* così come indicato dal documento apposito *Piano di qualifica V3.0*.

### 3.3.1 Misure di qualità

La qualità indica la capacità di superare processi di verifica e validazione con esiti positivi. La verifica indica la correttezza del processo, mentre la validazione certifica il raggiungimento dei requisiti.

Per mantenere l'oggettività nella misurazione della qualità di un ente si utilizzano delle metriche. Le metriche per valutare la qualità di prodotto e di processo sono caratterizzate da dei codici identificativi che seguono il seguente formato:

**M[Utilizzo][Numero identificativo]**

dove:

- M: indica una metrica di qualità;
- **Utilizzo:** se la metrica è per:
  - **PP:** processo primario,
  - **PS:** processo supporto,
  - **PO:** processo organizzativo,
  - **CD:** controllo documentazione,
  - **SW:** software,
- **Numero identificativo:** è un numero progressivo identificativo che viene incrementato ad ogni metrica individuata corrispondente a un determinato tipo.

### 3.3.2 Strumenti utilizzati

Nome	Descrizione	Riferimenti
Fogli <i>Google</i>	Fogli <i>Google</i> è un servizio sviluppato da <i>Google</i> , che permette di creare, modificare e condividere fogli di lavoro. È stato utilizzato per la creazione di grafici e tabelle organizzative.	<a href="http://www.google.it">www.google.it</a>
<i>LanguageTool</i>	<i>Software</i> che evidenzia errori grammaticali e logici effettuati durante la stesura di testo. Utilizzato in fase di stesura e verifica dei documenti.	<a href="http://languagetool.org">languagetool.org</a>

**Tabella 13:** Principali strumenti utilizzati nei processi di gestione della qualità.

## 3.4 Verifica

Il processo di verifica prende in ingresso un prodotto e lo ritorna corretto, coeso e completo. Per ottenere tali certificazioni ci si affida a sotto processi di analisi e test. In questa sezione verranno quindi descritti i procedimenti che il gruppo ha deciso di adottare per attuare il processo di verifica ai prodotti creati.

### 3.4.1 Verifica della documentazione

Il processo di verifica della documentazione viene fatto:

- Automaticamente: utilizzando strumenti di autocorrezione,
- Manualmente: ispezionando il testo e modificandone errori riscontrati. Il processo di ispezione può avvenire in due modalità:
  - Completa: viene ispezionato tutto il documento alla ricerca di eventuali errori, richiede l'utilizzo di più risorse,
  - Mirata: viene ispezionato solo una sezione del testo specifica, è da preferire alla modalità precedente in quanto utilizza meno risorse, ma richiede una comunicazione massima tra verificatore e redattore in quanto il primo deve essere informato della o delle sezioni modificate, così da concentrarsi su quelle. A questo scopo ritorna molto utile il Registro delle modifiche (3.1.5.3).

### 3.4.2 Verifica del codice

Il processo di verifica del codice utilizza due diverse tipologie di analisi:

- Statica: viene ispezionato il codice prodotto e viene valutato e all'occorrenza corretto sulla base di regole di programmazione preimpostati dal team,
- Dinamica: viene eseguito il programma e vengono ricercati errori o *bug*.

### 3.4.2.1 Test

Il *testing* del codice fa parte del processo della sua verifica e la sua esecuzione deve essere ripetibile. È quindi necessario rendere i test il più automatizzati possibile.

#### 3.4.2.1.1 Classificazione dei test

I *test* sono descritti nel documento *Piano di qualifica V3.0* e in esso sono classificati e descritti mediante codice identificativo.

Ogni test viene quindi identificato dall'espressione regolare:

**T[Tipologia test][ID Numerico]**

Dove:

- Tipologia Test: può assumere i valori riassunti nella seguente tabella

Sigla	Significato	Descrizione
I	Test di integrazione	Servono a testare che i moduli quando integrati funzionino come previsto, ovvero testare che i moduli che funzionano bene individualmente non abbiano problemi quando integrati
S	Test di sistema	Condotti su un sistema integrato completo, servono per valutare la conformità del sistema ai requisiti specificati
F	Test funzionali	I test funzionali sono procedure di controllo eseguite per verificare che un sistema o un'applicazione soddisfino correttamente le specifiche e le funzionalità richieste, garantendo il loro corretto funzionamento.
U	Test di unità	Condotti su una parte di codice, servono per verificare che tale parte funzioni correttamente

**Tabella 14:** Tipologie di test

- ID Numerico: codice numerico che identifica un test univocamente tra tutti quelli creati.

### 3.4.3 Strumenti utilizzati

Nome	Descrizione	Riferimenti
------	-------------	-------------

<i>Fogli Google</i>	Fogli <i>Google</i> è un servizio sviluppato da <i>Google</i> , che permette di creare, modificare e condividere fogli di lavoro. È stato utilizzato per la creazione di grafici e tabelle organizzative.	<a href="http://www.google.it">www.google.it</a>
<i>Cobertura</i>	<i>Cobertura</i> è uno strumento di <i>code coverage</i> per <i>Java</i> che traccia l'esecuzione dei test e fornisce informazioni sulla percentuale di codice testato. Genera <i>report</i> dettagliati che aiutano gli sviluppatori a identificare aree non coperte dai test e migliorare la qualità del <i>software</i> .	<a href="http://cobertura.github.io">cobertura.github.io</a>
<i>Istanbul</i>	Istanbul è uno strumento di <i>code coverage</i> per <i>JavaScript</i> che consente di misurare la quantità di codice testato dai test automatizzati. Supporta diversi <i>framework</i> e genera <i>report</i> dettagliati che aiutano gli sviluppatori a identificare le porzioni di codice non coperte dai test.	<a href="http://istanbul.js.org/">istanbul.js.org/</a>

**Tabella 15:** Principali strumenti utilizzati nei processi di gestione della qualità.

## 3.5 Validazione

I prodotti sviluppati durante il corso del progetto devono essere in linea con le richieste e i bisogni del committente, è necessario quindi un processo di validazione.

### 3.5.1 Attività

Il processo è interamente gestito dal responsabile di progetto che controllerà il risultato del processo di verifica e valuterà se:

- Accettare il risultato,
- Richiedere una nuova verifica esprimendo le nuove richieste al o ai verificatori.

## 4 Processi Organizzativi

Questa sezione descrive le modalità alle quali il *team* si atterrà per la gestione delle risorse (umane, temporali ed economiche).

### 4.1 Gestione dei processi

In questo processo rientrano tutte le attività inerenti la gestione dei processi. In particolare:

- Assegnazione dei ruoli e dei compiti,
- Pianificazione  $\text{sprint}_G$  temporale,
- Pianificazione e gestione del bilancio temporale ed economico,
- Revisione e valutazione periodica del lavoro svolto,
- Comunicazione con azienda proponente,
- Comunicazione tra i membri del gruppo,
- Gestione dei rischi,
- Produzione di documenti e software.

#### 4.1.1 Ruoli di progetto

##### 4.1.1.1 Elenco dei ruoli

- Responsabile di progetto: garantisce il regolare svolgimento delle attività pianificate, fornendo supporto quando necessario cercando di risolvere eventuali problemi. In particolare si occupa di:
  - Individuazione e assegnazione dei ruoli necessari per lo  $\text{sprint}_G$  del quale è responsabile,
  - Individuazione di eventuali problemi e supporto nella loro risoluzione,
  - Approvazione documentazione prodotta per il successivo rilascio,
  - Rappresentare il gruppo nelle comunicazioni con l'azienda.
- Amministratore: gestisce gli strumenti necessari all'ambiente di lavoro. In particolare:
  - Gestione di archiviazione e versionamento di *software* e documenti,
  - Gestione dell'efficienza dell'ambiente di sviluppo, fornendo strumenti adeguati ai membri del gruppo,
  - Redazione e manutenzione del Documento *Norme di progetto*,
  - Garante applicazione del *Piano di Progetto*.
- Analista: studia e documenta problematiche e vantaggi di servizi che il prodotto *software* deve fornire. In particolare:
  - Esegue lo studio di fattibilità di eventuali idee e tecnologie proposte corredate da documenti riassuntivi a uso interno,

- Si occupa di modellazione concettuale dei servizi offerti dal prodotto, partendo dall'analisi dei bisogni espliciti e impliciti,
  - Individua e categorizza i requisiti,
  - Redige il documento *Analisi dei requisiti*.
- Progettista: determina le scelte realizzative proposte dall'analista. In particolare:
    - Individua una soluzione a partire dai requisiti individuati in fase di analisi,
    - Definisce l'architettura del prodotto in modo tale da soddisfare tutti e solo i requisiti fissati,
    - Definisce la progettazione di dettaglio in modo tale da guidare e delimitare il lavoro di codifica.
  - Programmatore: scrive il codice richiesto dalle figure sopra descritte. In particolare:
    - Scrive e documenta un codice manutenibile e versionato,
    - Si assicura che il codice prodotto sia facile da verificare applicando modalità e utilizzando strumenti descritti nella sezione apposita di questo documento.
  - Verificatore: verifica documenti e *software* prodotti utilizzando come criterio di valutazione questo documento. In particolare:
    - Assicura la corretta applicazione delle istruzioni contenute nel documento *Norme di progetto V2.0*,
    - Segnala in tempi immediati eventuali errori e correzioni necessarie al resto del *team*.

#### 4.1.2 Rotazione dei ruoli

Per requisito di progetto, la rotazione dei ruoli è obbligatoria e deve essere equa, continua ed esente da conflitti d'interesse.

I ruoli vengono decisi da responsabile di progetto e cambiano ad ogni *sprint<sub>G</sub>*

Ogni componente di *Out Of Bounds* deve assumere tutti i ruoli e deve assumerli in distribuzione equa tra gli altri membri del gruppo. L'assunzione di alcuni ruoli non può essere consecutiva, in particolare:

- Un verificatore non può diventare responsabile, se lo *sprint<sub>G</sub>* successivo richiede l'approvazione di documenti verificati dalla stessa persona,
- Un analista o programmatore non può diventare verificatore se lo *sprint* successivo richiede la verifica del proprio lavoro.

#### 4.1.3 Comunicazione

##### 4.1.3.1 Testuale

- Interna: avviene quotidianamente tramite applicazioni *Telegram<sub>G</sub>* e *Discord<sub>G</sub>*. All'interno di quest'ultimo sono presenti canali specifici per comunicazioni importanti e per la condivisione di materiale ritenuto utile o interessante,
- Esterna: le comunicazioni con il rappresentante aziendale avvengono utilizzando la casella di posta elettronica del gruppo presente a piè di pagina sulla sinistra.

#### 4.1.3.2 Verbale

Con comunicazione verbale si intendono le riunioni ufficiali effettuate.

- Interna:
  - Convocazione: il responsabile del progetto indice gli incontri ufficiali, considerando le disponibilità dei componenti del gruppo espresse mediante comunicazione testuale,
  - Verbale di riunione: a rotazione un componente del gruppo si dedica alla stesura del verbale che viene poi approvato del responsabile direttamente,
  - Strumenti utilizzati: vengono usati i canali vocali di *Discord<sub>G</sub>*.
- Esterna:
  - Convocazione: alla necessità il responsabile richiede un incontro al rappresentate aziendale. Gli incontri vengono effettuati alla necessità di chiarimenti urgenti o per fornire aggiornamenti e richiedere pareri sul lavoro svolto,
  - Verbale di riunione: a rotazione un componente del gruppo si dedica alla stesura del verbale che viene poi approvato del responsabile direttamente,
  - Strumenti utilizzati: viene usata una stanza virtuale di *Zoom<sub>G</sub>* così come indicato dall'azienda durante la presentazione del capitolato d'appalto.

#### 4.1.4 Gestione dei rischi

L'individuazione dei rischi è compito del Responsabile che, una volta identificati, li rende noti ai componenti del gruppo tramite il documento *Piano di Progetto V1.0*, indicandoli nel seguente modo:

**R[Tipo].[X]**

- R: Indica che il codice si riferisce a un rischio,
- Tipo: indica la tipologia del rischio individuato e può assumere i seguenti valori:

Valore	Significato
T	Tecnologico
R	Requisito
O	Organizzativo
I	Interno

**Tabella 16:** Valori utilizzati nella codifica dei rischi

- X: Valore numerico che inizia dal valore uno e viene incrementato di un'unità a ogni rischio aggiunto.



## 4.2 Gestione dell'infrastruttura

Il processo di gestione dell'infrastruttura descrive l'infrastruttura utilizzata in tutti i restanti processi. L'infrastruttura include *hardware*, *software*, strumenti, tecniche, *standard* o altro.

### 4.2.1 Strumenti di coordinamento

- $GanttProject_G$ : Applicativo per la creazione di diagrammi di *Gantt*,

<https://www.ganttproject.biz/>.

- $Jira_G$ : come strumento collaborativo per il controllo gestionale, per il controllo della qualità e per il coordinamento delle attività.

<https://www.atlassian.com/software/jira>.

### 4.2.2 Strumenti di versionamento

Lo strumento utilizzato è  $GitHub_G$ , dove l'amministratore ha diverse  $repository_G$ . Tutti i componenti del gruppo hanno accesso a entrambe le  $repository_G$ .

## 4.3 Formazione

In questa sottosezione si definiscono le norme riguardanti il processo di formazione sugli strumenti utilizzati nel corso del progetto dei membri del gruppo.

### 4.3.1 Modalità di formazione

Il processo di formazione di ogni componente del gruppo avviene principalmente autonomamente, ma tutti i componenti del gruppo si sono resi disponibili a fornire supporto all'apprendimento di strumenti a loro familiari.

I nuovi strumenti introdotti dagli amministratori vengono seguiti invece dalla distribuzione di un breve documento che descrive le loro funzionalità.