Università degli Studi di Padova

Catch em All - *CAPTCHA: Umano o Sovraumano?*
Email: catchemallswe3@gmail.com

Norme di Progetto

Versione	(0.0.5)
Approvazione	(modifica)
Redazione	(Gabriele Da Re, Zhen Wei Zheng e Luca Brugnera)
Verifica	(modifica)
Stato	(In sviluppo)
Uso	(modifica)
Distribuzione	(modifica)

Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.0.5	26/11/2022	Completato sezione "Documentazione"	Zhen Wei Zheng	Amministratore
0.0.4	23/11/2022	Definita e implementata la sezione "Processi primari" del documento	Luca Brugnera, Gabriele Da Re	Amministratore, Amministratore
0.0.3	22/11/2022	Stesura sezione "Documentazione"	Zhen Wei Zheng	Amministratore
0.0.2	16/11/2022	Impostazione layout documento	Zhen Wei Zheng	Amministratore
0.0.1	15/11/2022	Creazione e prime definizioni del documento	Luca Brugnera	Amministratore

Indice

1 Introduzione

1.1 Scopo del documento

Dal proponente Zucchetti S.p.A. viene evidenziato, nel capitolato da loro proposto, una criticità negli attuali sistemi di sicurezza sulla rilevazione dei bot rispetto agli esseri umani. Oggi giorno il meccanismo più utilizzato per risolvere questo problema è il test CAPTCHA.
Un bot non è altro che una procedura automatizzata che, in questo caso, ha fini malevoli, come per esempio:

- registrazione presso siti web;
- creazione di spam;
- violare sistemi di sicurezza;

I bot, grazie alle nuove tecnologie sviluppate con sistemi che utilizzano principalmente l'intelligenza artificiale, riescono a svolgere compiti che fino a poco tempo fa venivano considerati impossibili da svolgere per una macchina.
Ciò evidenzia che i CAPTCHA attuali risultano sempre più obsoleti, non andando a individuare correttamente tutti i bot, se non quasi nessuno.
Un'altra criticità individuata dal proponente è il sistema di classificazione delle immagini che sta effettuando Google grazie al proprio reCAPTCHA, che attualmente è il sistema più diffuso.
Questa criticità nasce dal beneficio che questa big tech ottiene dall'interazione degli utenti nel risolvere le task proposte, che portano alla creazione di enormi dataset di immagini classificate che possono essere utilizzate per l'apprendimento dei propri sistemi di machine learning o vendibili a terzi.
Il capitolato C1 richiede di sviluppare una applicazione web costituita da una pagina di login provvista di questo sistema di rilevazione in grado di distinguere un utente umano da un bot.
L'utente quindi, dopo aver compilato il form in cui inserirà il nome utente e la password, dovrà svolgere una task che sarà il cosiddetto test CAPTCHA.

1.2 Scopo del prodotto

Il prodotto come finalità prevede il raggiungimento di un buon grado di distinzione tra bot ed esseri umani attraverso l'implementazione di un captcha il quale sfrutta varie tecniche e metodologie per il raggiungimento di tale scopo.

1.3 Glossario

Per evitare ambiguità relative al linguaggio utilizzato nei documenti prodotti, viene fornito il **Glossario v1.0.0** (per ora teorica la sua versione **DA MODIFICARE)**. In questo documento sono contenuti tutti i termini tecnici, i quali avranno una definizione specifica per comprenderne al meglio il loro significato.

1.4 Riferimenti

1.4.1 Riferimenti normativi

Link vari da aggiungere in seguito

1.4.2 Riferimenti informativi

- Capitolato C1 "CAPTCHA: umano o sovrumano?" `https://www.math.unipd.it/~tullio/IS-1/2022/Progetto/C1.pdf`
- Slide T06 del corso di Ingegneria del Software – Analisi dei requisiti: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T06.pdf`
- Slide P03 del corso di Ingegneria del Software – Diagrammi dei casi d'uso: `https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20Use%20Case.pdf`;
- Regolamento del progetto didattico – Materiale didattico del corso di Ingegneria del Software:: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/PD02.pdf`;

2 Processi primari

2.1 Acquisizione

Zucchetti S.p.A. richiede la realizzazione di un progetto creativo riguardante lo sviluppo di un sistema Captcha attraverso l'esposizione della lettera di presentazione *"CAPTCHA: Umano o Sovrumano?"* in data 18 ottobre 2022.

2.2 Fornitura

Successivamente alla presentazione dei capitolati *CatchEmAll* si riunisce per valutare le proposte e le opinioni dei componenti del team: attraverso un processo di valutazione (inizialmente generico poi specifico, riassunto nella sezione *Motivazione scelta capitolato* del documento lettera di presentazione) emerge una preferenza per il progetto proposto dal referente Dr. Gregorio Piccoli.
In seguito viene schedulata una riunione con il proponente con l'obiettivo di approfondire e consolidare le richieste.
In data 28 ottobre 2022 viene inviata al committente la *lettera di presentazione* il quale, in seguito ad una richiesta di modifica delle tempistiche di consegna, ci aggiudica l'appalto.

2.3 Sviluppo

2.3.1 Analisi dei requisiti

Il documento *Analisi dei requisiti* contiene in maniera formale:

- descrizione generale del prodotto
- casi d'uso
- requisiti, i quali vengono estrapolati da:
  - documento di presentazione del capitolato
  - confronto tra il gruppo e con il proponente

Struttura dei requisiti : ogni requisito è identificato da un codice univoco così composto:
`R<<tipologia di requisito>>-<<id>>`
Dove:

- «tipologia di requisito» identifica la classe tra le seguenti:
  - funzionale
  - qualità
  - vincolo
- «id» identifica il requisito nella classe di appartenza

Nel documento vengono raggruppati per categoria specificandone:

- grado di obbligatorietà

- descrizione
- fonte

Struttura dei casi d'uso : ogni caso d'uso è identificato utilizzando la seguente convenzione di nomenclatura:
`UC<<id>>.<<progressivo>>`
Dove

- id identifica l'use classe
- progressivo identifica eventuali sottocasi

2.3.2 Versionamento

GitHub è lo strumento utilizzato dal gruppo per il versionamento del codice.
Il team è identificato in tale piattaforma come organizzazione (vedi). Inoltre, al fine di documentare il più possibile, ogni commit che porta valore al progetto contiene il riferimento al ticket che completa (totalmente o anche solo parzialmente).

2.3.3 GitHub Workflow

Tutti i titoli e le descrizioni dei commit devono essere fatti in inglese per conformità tra essi.
Il Workflow viene gestito concorrentemente da GitHub e JIRA, in JIRA vengono create ed organizzate le issue, una volta fatto ciò si procede attraverso github alla creazione del branch relativo alla issue da risolvere.
Tale ramo ha nome codificato come:

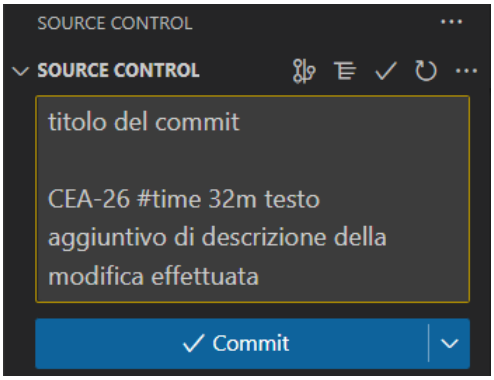
CEA–num–titolo–della–issue

Questo permette di identificare titolo e numero della issue di appartenenza. Una volta fatto ciò viene creato un **primo commit**, nel cui messaggio è specificata l'avvenuta presa in carico della issue, la quale dovrà passare dallo stato "to do" allo stato "in progress"
Ciò è garantito dal suddetto commit message contenente la stringa:

CEA–num #open <testo aggiuntivo>

Una volta fatto ciò è possibile lavorare liberamente sul proprio ramo di feature.
Ad ogni aggiornamento dell'attività svolta si dovrà fare riferimento alla issue e specificare il tempo impiegato per lo svolgimento di tale attività includendo nella descrizione:

- Visual Studio Code:



- Git Bash:

```
git commit –m "titolo del commit" –m "CEA–26 #time 32m ..."
```

Comando generico da aggiungere nel corpo del messaggio, non nel titolo del commit:

```
CEA–num #time ww dd hh mm <testo aggiuntivo>
```

Così facendo è permesso specificare a scelta settimane, giorni, ore e minuti di lavoro, ad esempio:

```
CEA–26 #time 1h aggiunto github Workflow
```

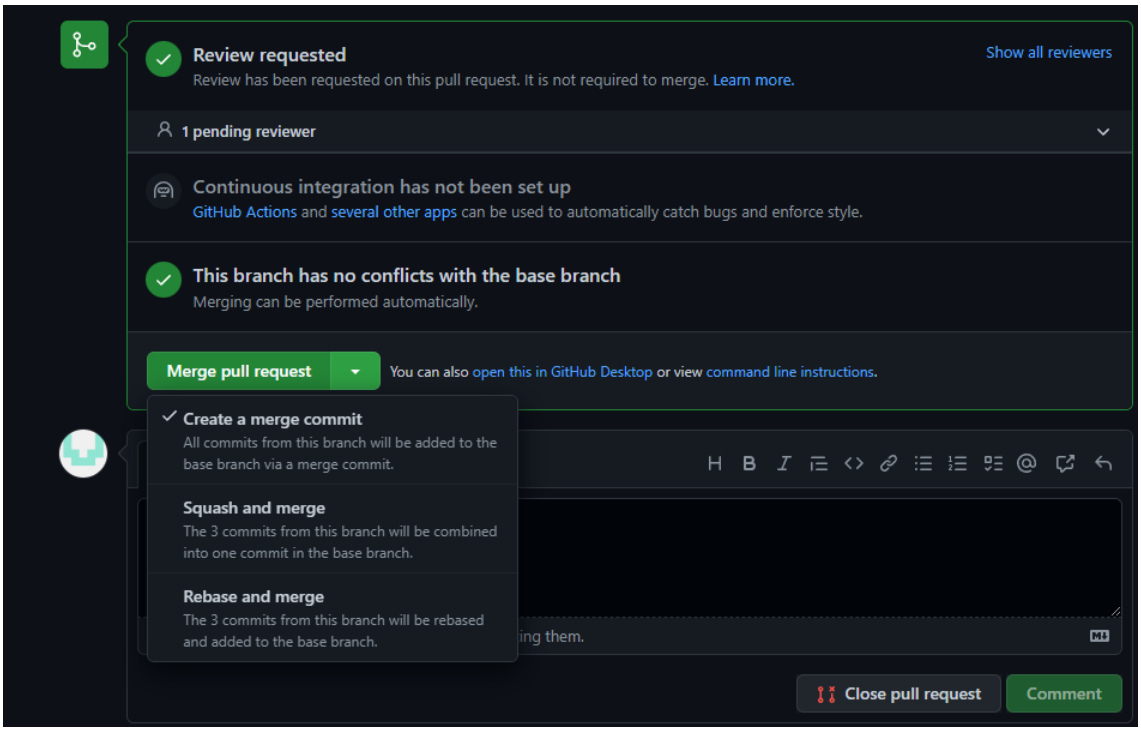
Ciò aggiunge 1h alle ore di lavoro impiegate per la issue con ID CEA-26, e come testo aggiuntivo per il commit "aggiunto github Workflow", ignorato da JIRA.
Una volta terminata l'attività, sarà necessario passare allo stato di revisione, il quale permette di verificare il corretto svolgimento del compito eseguito. Questo è permesso da un ultimo commit prima della revisione, con messaggio da includere nella descrizione(non titolo):

```
CEA–num #review #time ww dd hh mm <testo aggiuntivo>
```

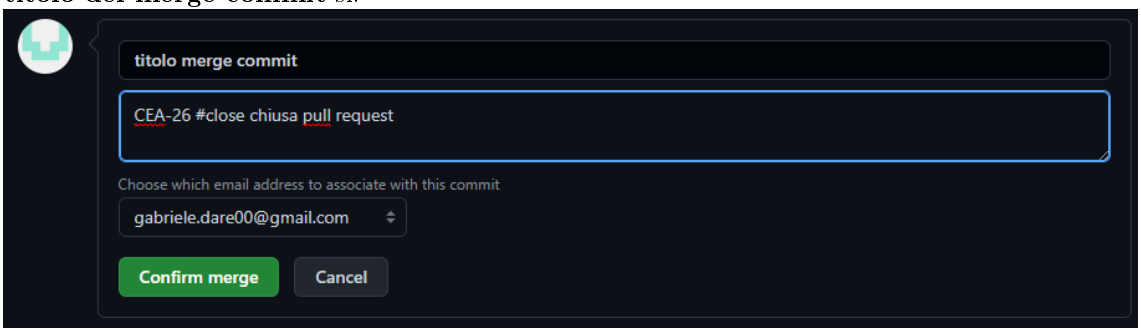
Questo permette lo spostamento della issue dallo stato "in progress" allo stato "in review".
Per permettere la revisione è necessario aprire una pull request, il titolo deve corrispondere al nome del branch. Una volta revisionata la issue, se presenta qualche problema può essere spostata allo stato "in progress" dal pannello JIRA. Altrimenti attraverso una pull request nel ramo "main" e con il seguente comando posto **nel TITOLO del commit di chiusura della pull request** la issue verrà chiusa e considerata completata:

```
CEA–num #close <testo aggiuntivo>
```

Una volta chiusa, sempre dalla pull request su github, **si elimina il ramo di feature** creato precedentemente.



La descrizione è a titolo esemplificativo, il contenuto non influenza gli smart commit di JIRA, il **titolo del merge commit** si.



2.3.4 Issues tracking

JIRA, piattaforma che offre un servizio di *Issue Tracking* è il supporto scelto vista la qualità ed il numero dei servizi e delle estensioni che offre.
La definizione dei ticket è regolata dalla seguente convenzione:

- titolo e descrizione devono, oltre ad essere sempre presenti, eplicitare in maniera chiara il problema
- utilizzo di label
- stima del lavoro necessario al completamento
- corretto utilizzo di ereditarietà (rapporti di parentela)

Si è deciso di di adottare il framework **Scrum** per la gestione del ciclo di sviluppo del progetto con le seguenti caratteristiche:

- sprint della durata di una settimana
- utilizzo di una board avente 4 stati:
  - to do
  - in progress
  - in review (ogni ticket deve essere validato da uno o più componenti del gruppo per essere considerato chiuso)
  - done

JIRA dispone di un'integrazione con github che fornisce un meccanismo chiamato *smart commit* il quale permette la transizione dei ticket da uno stato ad un'altro attraverso comandi posti nei commit stessi, la sintassi utilizzata è la seguente

CEA–number #command <message body describing the commit>

Tra i comandi troviamo:

- **open**: permette di spostarsi da una issue nello stadio "to do" oppure "in review" allo stadio "in progress"
- **review**: permette lo spostamento della issue dallo stadio "in progress" oppure "done" allo stadio "in review"
- **close**: premette di spostarsi dallo stadio "in review" allo stadio "done"
- **close-no-rev**: permette in casi eccezionali di passare direttamente dallo stadio "in progress" allo stadio "done"

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo di questa sezione è definire gli standard neccessari per la stesura di tutti i documenti del progetto.

3.1.2 Ciclo di vita del documento

Tutti i documenti prodotti dal team seguono la seguente ciclo di vita:

- **Stesura**: il documento viene scritto utilizzando il metodo AGILE, adottando gli sprint di durata una settimana;
- **Revisione**:a fine di ogni sprint i documenti vengono revisionati da una persona diversa dal redattore. Solo dopo la revisione le modifiche e i nuovi contenuti scritti possono essere integrati nel documento;
- **Verifica**:avviene in un periodo specificato nel piano di progetto, tale attività viene svolto da almeno una persona. Il documento è considerato verificato quando i Verificatori dichiarano che le modifiche neccessarie sono portate a termine;
- **Approvazione**:il Responsabile di Progetto dichiara che il documento è completo in ogni sua parte e pronto per essere rilasciato, marcandolo come approvato;

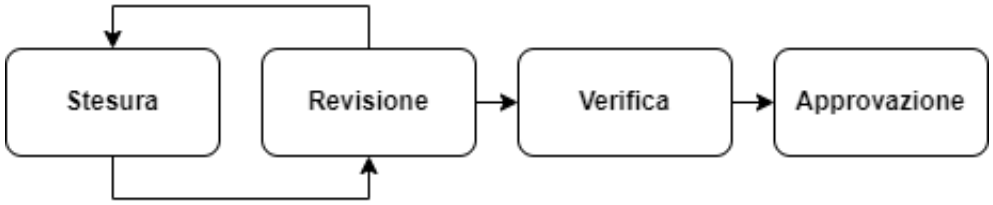


Figura 1: Ciclo di vita dei documenti

3.1.3 Struttura dei documenti

Tutti i documenti ufficiali seguono una struttura ben definita cosi da mantenere l'omogeneità. Più precisamente ogni documento è formato da:

- **Fontespizio**;
- **Registro delle modifiche**;
- **Indice**;
- **Contenuto principale**.

3.1.3.1 Fontespizio

Rappresenta la pagina iniziale del documento ed è strutturato come segue:

- **Logo dell'università**: logo dell'*Università di Padova* posizionato in centro alto della pagina, seguito dalla nomenclatura "Università degli Studi di Padova";
- **Logo del gruppo**: logo del gruppo, posizionato in centro, subito dopo la nomeclatura dell'università;
- **Nome del gruppo e del progetto**: il nome del gruppo e del capitolato scelto, seguito da un recapito email;
- **Nome del documento**: è il titolo del documento, definito in grassetto e posizionato al centro della pagina;
- **Tabella di descrizione**: è la tabella contenente le informazioni generali del documento.

3.1.3.2 Registro delle modifiche

I documenti che sono soggetti a modifiche periodiche sono dotati di un registro che ne memorizza lo storico. Il registro è fomato cosi:

- **Versione**: indica la versione del documento dopo la modifica;
- **Descrizione**: descrive brevemente la modifica apportata;
- **Data**: indica la data in cui è stata modificata il documento.

3.1.3.3 Indice

Per agevolare la lettura, tutti i documenti sono dotati di un indice. Le sezioni sono rappresentati da un numero seguiti dal titolo della sezione, ogni sottosezione deve riportare il numero della sezione madre e poi il numero proprio. I numeri devono partire dall'1.

3.1.3.4 Contenuto principale

La pagina del contenuto è costituita da:

- **Intestazione**: in alto a sinistra deve esserci il nome del gruppo *Catch em All*, in altro a destra si trova il numero e nome della sezione in cui si trova;
- **Pie di pagina**: in basso sinistra si trova il nome del documento e la sua versione attuale, in basso a destra viene indicato il numero della pagina in cui si trova e il numero di pagine complessive del documento.

3.1.4 Classificazione dei documenti

Tutti i documenti prodotti sono divisi in uso interno e uso esterno:

- **Uso interno**: sono documenti finalizzati a un uso interno al gruppo, tra cui *Norme di progetto* e *Verbali interni*;
- **Uso esterno**: sono documenti di interesse a tutti gli stakeholder, tra cui *Analisi dei requisiti*, *Verbali esterni*, *Piano di progetto*, (da completare).

3.1.5 Norme tipografiche

3.1.5.1 Nome del file

Di seguito viene descritto il formato dei nomi dei documenti:

- Iniziano tutti con la lettera minuscola;
- Se il nome comprende più parole allora ognuna di esse è separata dal simbolo '_';
- Deve essere seguito da un indicazione della propria versione.

La sigla della versione deve essere così strutturata:

v.X.Y.Z

- **X** indicato da un numero che parte da 0, corrisponde al numero di approvazioni del documento da parte del responsabile;
- **Y** indicato da un numero che parte da 0, corrisponde al numero di verifiche del documento da parte del verificatore, viene portato a 0 ad ogni incremento di **X**;
- **Z** indicato da un numero che parte da 0, corrisponde al numero di modifiche del documento da parte del redattore, viene portato a 0 ad ogni incremento di **X** e **Y**.

Esempi corretti: introduzione_v0.0.1; norme_di_progetto_v.0.0.1 .
Esempi non corretti: Norme_di_Progetto; NormeDiProgetto.

I verbali non seguono questa norma e hanno una nomenclatura diversa, poichè non subiscono variazioni dopo la prima redazione e hanno il seguente formato: (DA DEFINIRE)

3.1.5.2 Stile di testo

Di seguito vengono riportati i vari stili del testo e i loro usi:

- **Grassetto**: viene utilizzato per i termini negli elenchi puntati e per i titoli delle sezioni;
- **Corsivo**: viene utilizzato per il nome del gruppo, l'email del gruppo e il nome del progetto;
- **Link**: i link sono collegamenti esterni del documento.

3.1.5.3 Glossario

Le norme relative al *Glossario* sono:

- Ogni parola presente nel *glossario* viene contrassegnata con una 'G' a pedice;
- Se un termine compare nella sua stessa definizione all'interno del *glossario* esso viene contrassegnato.

3.1.5.4 Elenchi puntati e numerati

Di seguito vengono descritti come vengono utilizzati gli elenchi puntati e numerati:

- Ogni punto dell'elenco inizia con la lettera maiuscola;
- Alla fine di ogni punto vi è un ';';
- Dopo l'ultima voce vi è un '.';
- Se vi è un concetto da spiegare esso viene scritto in grassetto seguito da ':' e segue la spiegazione di esso.

3.1.5.5 Sigle TODO

Di seguito viene elencata una lista di sigle le quali si possono trovare nei documenti e i loro significati:

- **Documentazione**:
  - **AdR**: indica l' *Analisi Dei Requisiti*;
  - **NdP**: indica le *Norme Di Progetto*;
  - **PdP**: indica il *Piano Di Progetto*;
  - **PdQ**: indica il *Piano Di Qualifica*;
  - **MU**: indica il *Manuale Utente*;
  - **MdM**: indica il *Manuale del Manutentore*.
- **Ruoli**:
  - **Re**: indica il ruolo di *Responsabile di Progetto*;
  - **Am**: indica il ruolo di *Amministratore di Progetto*;
  - **An**: indica il ruolo di *Analista*;
  - **Pt**: indica il ruolo di *Progettista*;
  - **Pr**: indica il ruolo di *Programmatore*;
  - **Ve**: indica il ruolo di *Verificatore*.

3.1.5.6 Formato della data

Il team ha adottato il seguente formato per la data:

DD-MM-YYYY

Dove **DD** indica il giorno, **MM** indica il mese, **YYYY** indica l'anno.

3.1.6 Elementi grafici

3.1.6.1 Tabelle TODO

Con eccezzione per le tabelle delle modifiche, tutte le altre tabelle di ogni documento:

- Sono centrate orizzontalmente;
- Devono essere accompagnate da una didascalia che indichi il numero dell'immagine all'interno del documento e con una breve descrizione.

3.1.6.2 Immagini TODO

Le immagini sono anch'esse centrate orizzontalmente e devono avere una didascalia che indichi il numero dell'immagine all'interno del documento e con una breve descrizione.

3.1.7 Strumenti TODO

Di seguito vengono elencati gli strumenti usati per stendere i documenti:

- **LaTeX**: per la produzione dei documenti, il team ha deciso di usare il linguaggio di marckup *LaTeX*;
- **Microsoft Word**: per la stesura delle bozze;
- **StarUML**: per produrre diagrammi.

3.2 Gestione della configurazione

3.2.1 Scopo

Lo scopo di questa sezione è descrivere come il team ha deciso di mantenere traccia le varie documentazioni stese.

3.2.2 Sistemi software utilizzati TODO

Per gestire i file e gli aggiornamenti dei documenti, viene utilizzato il servizio offerto da Github.

3.2.3 Struttura del repository TODO

Di seguito viene dato una lista di repository presente su Git

- **catchEmAll-SWE/catchEmAll-Docs** è il repository contenente documentazione riguardante:
  - Assegnazione dell'appalto (lettera di candidatura);
  - Diario di bordo;
  - Ricerche e documentazione prodotta dal team;
  - Specifiche tecniche del software;
  - Link dei verbali (interni ed esterni).
- **catchEmAll-SWE/catchEmAll-Code(possibile repo)**

Confluence (strumento JIRA) contiene invece i verbali e i documenti retrospettivi: tale sceltà è stata guidata dalla presenza in questo strumento di template, i quali ne facilitano la scrittura.

3.2.4 Gestione delle modifiche

Per evitare i conflitti tra le modifiche e mantenere in ordine i file, ogni volta che viene apportata una modifica deve prima essere caricata su un branch per essere revisionata dal Ve poi viene mergiata nel branch principale **DA RISCRIVERE MEGLIO**

3.2.5 Tipi di file presenti

Nella repository *catchEmAll-Docs* sono presenti esclusivamente file **.tex**, **.png** e /**pdf**. Altri file prodotte durante la stesura dei documenti di esetensione diversa da quelle citate sono escluse attraverso il file **.gitignore**.

4 Processi organizzativi

4.1 Gestione delle comunicazioni

4.1.1 Comunicazioni interne

Le comunicazioni interne:

- riguardano solamente i componenti del team
- avvengono su *WhatsApp*.
- utilizzate per:
  - comunicazioni istantanee tra tutti i componenti
  - discussioni
  - pianificazione degli incontri
  - *daily scrum meeting*

4.1.2 Comunicazioni esterne

Le comunicazioni esterne:

- riguardano il gruppo e le altre figure (proponente e committente)
- utilizzo del dominio di gruppo (*catchemallswe3@gmail.com*) di posta elettronica
- utilizzate per comunicazioni ufficiali tra il team e le altre figure

4.2 Gestione degli incontri

4.2.1 Incontri interni

Gli incontri interni sono necessari sia per una corretta adozione del framework Scrum (incontro organizzativo settimanale) sia per permettere al team di interagire direttamente, discutendo, proponendo e valutando idee, problematiche e possibili soluzioni: per questo si tratta di uno strumento largamente utilizzato
Si predilige la modalità virtuale per comodità cercando di schedulare riunioni in cui tutti riescano a partecipare.
La piattaforma utilizzata è *discord*, la quale permette la creazione e l'utilizzo di:

- canali testuali
- canali video (con possibilità di condivisione schermo)

Al termine degli incontri il responsabile di progetto inserisce nello sprint corrente il compito di redigere i verbali.

4.2.2 Incontri esterni

Gli incontri esterni sono schedulati in seguito alla presenza di dubbi (implementativi, riguardanti requisiti o richieste di altro tipo) all'interno del team: questi incontri sono preceduti dallo svolgimento di una o più riunioni interne nelle quali si affrontano e si definiscono tali problematiche.
Per quanto riguarda l'organizzazione viene contattato tramite email il referente di progetto proponendogli diverse date e orari affinchè si trovi quella più comoda per entrambe le parti.
Come per quelli interni gli incontri esterni sono tenuti in modalità virtuale ma a loro differenza si utilizza una riunione *Zoom* definita dal gruppo.
I verbali hanno lo scopo di documentare in maniera dettagliata tutti gli argomenti trattati affinchè si possa costruire uno storico identificando e motivando le decisioni prese.
Come per quelli interni il responsabile di progetto inserisce nello sprint corrente il compito di redigere tali documenti.