**Università degli Studi di Padova**


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

# Norme di Progetto

| | |
|---|---|
| **Versione** | (0.0.2) |
| **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 | Descrizione | Data |
|---|---|---|
| 0.0.2 | Impostazione del layout | 16/11/2022 |
| 0.0.1 | Creazione e prime definizioni del documento | 15/11/2022 |

# Indice

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di mantener traccia del *way of working* e delle *best practices* che il team si impegna a rispettare durante l'intero svolgimento del progetto "*CAPTCHA: Umano o Sovrumano?*".

## 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.

# 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

**2.3.1.1 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

**2.3.1.2 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>

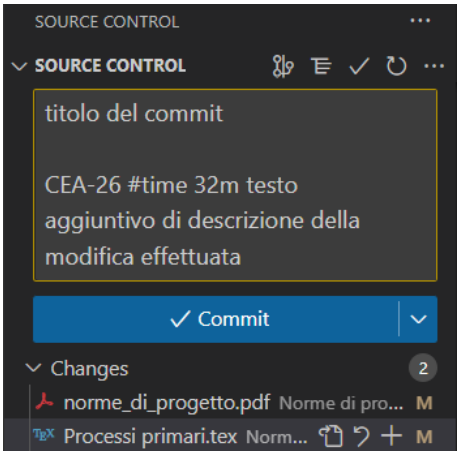
Una volta fatto ciò è possibile lavorare liberamente sul proprio ramo di feature.
**Ad ogni** aggiornamento dell'attività svolta si dovrà fare riferimento alla issue e specificare il tempo impiegato per lo svolgimento di tale attività includendo nella descrizione:

- 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:

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
```

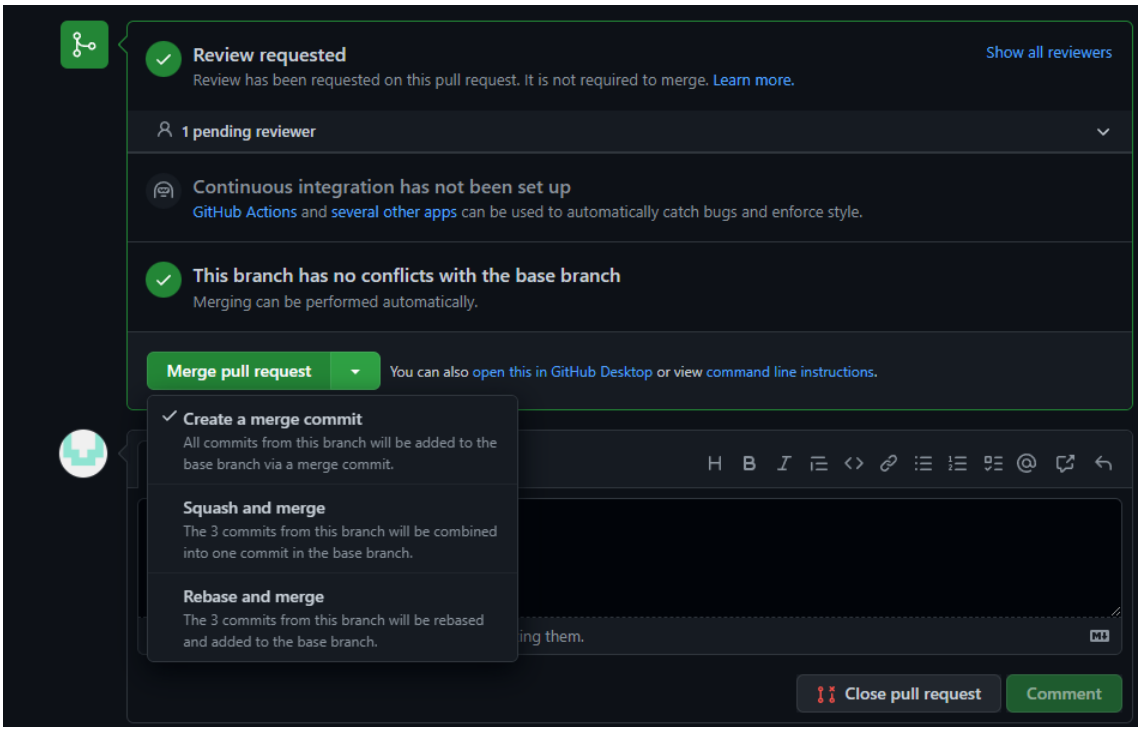
Ciò aggiunge 1h alle ore di lavoro impiegate per la issue con ID CEA-26, e come testo aggiuntivo per il commit "aggiunto github Workflow", ignorato da JIRA.
Una volta terminata l'attività, sarà necessario passare allo stato di revisione, il quale permette di verificare il corretto svolgimento del compito 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>
```

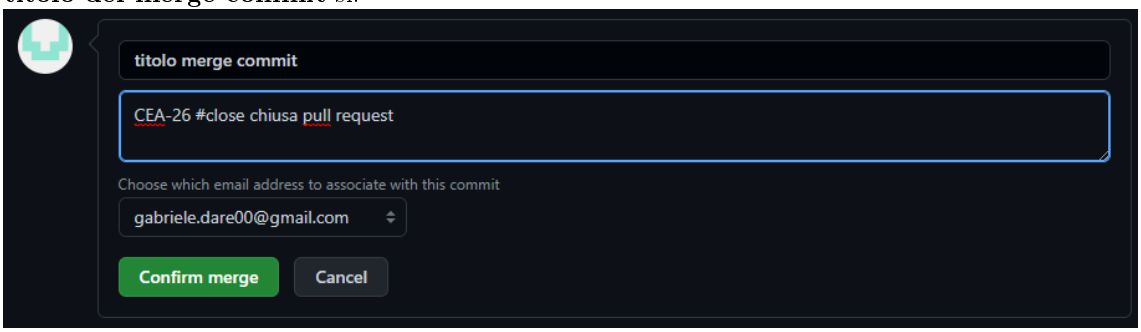
Questo permette lo spostamento della issue dallo stato "in progress" allo stato "in review".
Per permettere la revisione è necessario aprire una pull request, il titolo deve corrispondere al nome del branch. Una volta revisionata la issue, se presenta qualche problema può essere spostata allo stato "in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

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

Viene chiusa. Una volta chiusa sempre dalla pul 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

GitHub dispone di un repository contente documentazione riguardante:

- assegnazione appalto (lettera di candidatura)
- diario di bordo
- ricerche e documentazione prodotta dal team
- specifiche tecniche del software
- link ai verbali (interni ed esterni)

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 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**.

### 3.2.1 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 il nome del progetto in questione, seguito da un recapito email;
- **Nome del documento**: è il titolo del documento, in grassetto e posizionato al centro della pagina;
- **Tabella di descrizione**: è la tabella contenente le informazioni generali del documento.

### 3.2.2 Registro delle modifiche

I documenti che sono soggetti alle modifiche continue periodiche sono dotati di un registro che li memorizza. 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;

### 3.2.3 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.

### 3.2.4 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 progetto 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.

## 3.3 Classificazione dei documenti

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

- **Uso interno**: sono documenti usati esclusivamente dal gruppo, tra cui *Norme di progetto* e *Verbali interni*;
- **Uso esterno**: sono documenti per i componenti fuori dal gruppo, tra cui *Analisi dei requisiti*, *Verbali esterni*, *Piano di progetto*, (da completare);

# 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.