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.

1.3 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.

1.4 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.

1.5 Sviluppo

1.5.1 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).

1.5.2 GitHub Workflow

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 èviene 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. Ogni aggiornamento dell'attività svolta dovrà fare riferimento alla issue e specificare il tempo impiegato per lo svolgimento di tale attività:

```
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:

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

Questo permette lo spostamento della issue dallo stato "in progress" allo stato "in review".
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 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.

1.5.3 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"

2 Processi di supporto

2.1 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