



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	Data	Descrizione	Autore	Ruolo
0.0.4	26/11/2022	Completato "Documentazione"	Zhen Wei Zheng	Amministratore
0.0.3	22/11/2022	Stesura "Documentazione"	Zhen Wei Zheng	Amministratore
0.0.2	16/11/2022	Impostazione del layout	Zhen Wei Zheng	Amministratore
0.0.1	15/11/2022	Creazione e prime definizioni del documento	Luca Bruniera	Amministratore

Indice

1	Introduzione	2
1.1	Scopo del documento	2
1.2	Scopo del prodotto	2
2	Processi primari	3
2.1	Acquisizione	3
2.2	Fornitura	3
2.3	Sviluppo	3
2.3.1	Versionamento	3
2.3.2	GitHub Workflow	3
2.3.3	Issues tracking	4
3	Processi di supporto	6
3.1	Documentazione	6
3.1.1	Struttura dei documenti	6
3.1.1.1	Fontespizio	6
3.1.1.2	Registro delle modifiche	7
3.1.1.3	Indice	7
3.1.1.4	Contenuto principale	7
3.1.2	Classificazione dei documenti	7
3.1.3	Norme tipografiche	7
3.1.3.1	Nome del file	7
3.1.3.2	Stile di testo	8
3.1.3.3	Glossario	8
3.1.3.4	Elenchi puntati e numerati	8
3.1.3.5	Sigle TODO	9
3.1.3.6	Formato della data	9
3.1.4	Elementi grafici	9
3.1.4.1	Tabelle TODO	9
3.1.4.2	Immagini TODO	9
3.1.5	Strumenti TODO	10
4	Processi organizzativi	11
4.1	Gestione delle comunicazioni	11
4.1.1	Comunicazioni interne	11
4.1.2	Comunicazioni esterne	11
4.2	Gestione degli incontri	11
4.2.1	Incontri interni	11
4.2.2	Incontri esterni	12

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 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).

2.3.2 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 è 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. 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 pull request su github, si elimina il ramo di feature creato precedentemente.

2.3.3 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, explicitare 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 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:** permette 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 contenente 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 scelta è stata guidata dalla presenza in questo strumento di template, i quali ne facilitano la scrittura

3.1.1 Struttura dei documenti

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

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

3.1.1.1 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 nomenclatura 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.

3.1.1.2 Registro delle modifiche

I documenti che sono soggetti alle modifiche continue periodiche sono dotati di un registro che li memorizza. Il registro è formato così:

- **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;

3.1.1.3 Indice

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

3.1.1.4 Contenuto principale

La pagina del contenuto è costituita da:

- **Intestazione:** in alto a sinistra deve esserci il nome del gruppo *Catch em All*, in alto 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.

3.1.2 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);

3.1.3 Norme tipografiche

3.1.3.1 Nome del file

Di seguito viene descritto il formato dei nomi dei file:

- I nomi dei file iniziano tutti con la lettera minuscola;
- Se il nome comprende più parole allora ognuna di esse è separata dal simbolo '_' ;
- Il nome deve essere seguita 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 volte di approvazione del documento da parte del responsabile;
- **Y** indicato da un numero che parte da 0, corrisponde al numero di volte di verifica 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 volte 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. Poiche non subiscono variazioni dopo la prima redazione, hanno il seguente formato: (DA DEFINIRE)

3.1.3.2 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, il nome del progetto;
- **Link**: i link sono collegamenti esterni del documento;

3.1.3.3 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.

3.1.3.4 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.

3.1.3.5 Sigle TODO

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

- **Ruoli:**

- **Re:** indica il *Analisi Dei Requisiti*;
- **Am:** indica l'*Norme Di Progetto*;
- **An:** indica l'*Piano Di Progetto*;
- **Pt:** indica il *Piano Di Qualifica*;
- **Pr:** indica il *Manuale Utente*;
- **Ve:** indica il *Manuale del Manutentore*.

- **Documentazione:**

- **AdR:** indica il documento *Responsabile di Progetto*;
- **NdP:** indica il documento *Amministratore di Progetto*;
- **PdP:** indica il documento *Analista*;
- **PdQ:** indica il documento *Progettista*;
- **MU:** indica il documento *Programmatore*;
- **MdM:** indica il documento *Verificatore*.

3.1.3.6 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.

3.1.4 Elementi grafici

3.1.4.1 Tabelle TODO

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

- Sono centrati orizzontalmente:
- Devono essere accompagnati da una didascalia che indichi il numero dell'immagine all'interno del documento e una breve descrizione.

3.1.4.2 Immagini TODO

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

3.1.5 Strumenti TODO

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

- **L^AT_EX**: per la produzione dei documenti, il team ha deciso di usare il linguaggio di markup *L^AT_EX*;
- **Microsoft Word**: per la stesura delle bozze;
- altro

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 (*catchemallsw3@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 affinché 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 affinché 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.