



Università degli Studi di Padova



Catch em All - *CAPTCHA: Umano o Sovraumano?*

Email: catchemallswe3@gmail.com

Norme di Progetto

Versione	(0.0.6)
Approvazione	(modifica)
Redazione	(Gabriele Da Re, Zhen Wei Zheng, Matteo Stocco e Luca Brugnera)
Verifica	(modifica)
Stato	(In sviluppo)
Uso	(modifica)
Distribuzione	(modifica)

Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.0.6	24/12/2022	Completato sezione "Processi primari"	Matteo Stocco	Amministratore
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	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	5
1.4.1	Riferimenti normativi	5
1.4.2	Riferimenti informativi	5
2	Processi primari	6
2.1	Acquisizione	6
2.2	Fornitura	6
2.2.1	Scopo	6
2.2.2	Rapporto con il proponente	6
2.2.3	Prodotti e documenti forniti	6
2.2.3.1	Documenti per la candidatura	6
2.2.3.2	Analisi _dei_ requisiti v 1.0.0	7
2.2.3.3	Piano _di_ progetto v 1.0.0	7
2.2.3.4	Piano _di_ qualifica v 1.0.0	9
2.2.3.5	Proof of Concept	10
2.3	Sviluppo	10
2.3.1	Scopo	10
2.3.2	Analisi dei requisiti	10
2.3.2.1	Scopo e descrizione	10
2.3.2.2	Struttura dei casi d'uso	11
2.3.2.3	Struttura dei requisiti	11
2.3.3	Progettazione	12
2.3.3.1	Scopo e descrizione	12
2.3.3.2	Suddivisione attività	12
2.3.3.3	Proof of concept	12
2.3.3.4	Progettazione architettuale	13
2.3.3.5	Progettazione di dettaglio	13
2.3.4	Codifica	13
2.3.4.1	Scopo e descrizione	13
2.3.4.2	Suddivisione attività	13
3	Processi di supporto	14
3.1	Documentazione	14
3.1.1	Scopo	14
3.1.2	Ciclo di vita del documento	14
3.1.3	Struttura dei documenti	14
3.1.3.1	Fontespizio	15
3.1.3.2	Registro delle modifiche	15
3.1.3.3	Indice	15

3.1.3.4	Contenuto principale	15
3.1.4	Classificazione dei documenti	15
3.1.5	Norme tipografiche	16
3.1.5.1	Nome del file	16
3.1.5.2	Stile di testo	16
3.1.5.3	Glossario	16
3.1.5.4	Elenchi puntati e numerati	17
3.1.5.5	Sigle TODO	17
3.1.5.6	Formato della data	17
3.1.6	Elementi grafici	17
3.1.6.1	Tabelle TODO	17
3.1.6.2	Immagini TODO	18
3.1.7	Strumenti TODO	18
3.2	Gestione della configurazione	18
3.2.1	Scopo	18
3.2.2	Sistemi software utilizzati TODO	18
3.2.3	Struttura del repository TODO	18
3.2.4	Gestione delle modifiche	18
3.2.5	Tipi di file presenti	19
3.3	Assicurazione della qualità	19
3.3.1	Scopo	19
3.3.2	Denominazione obiettivi di qualità	19
3.3.3	Denominazione metriche di qualità	19
3.4	Verifica	20
3.4.1	Scopo	20
3.4.2	Analisi statica	20
3.4.2.1	Walkthrough	20
3.4.2.2	Inspection	21
3.4.3	Analisi dinamica	21
3.4.4	Denominazione test di verifica	21
3.5	Validazione e collaudo	22
3.5.1	Scopo	22
3.6	Usabilità	22
3.6.1	Scopo	22
4	Processi organizzativi	23
4.1	Gestione progetto	23
4.1.1	Scopo	23
4.1.2	Gestione ruoli	23
4.1.2.1	Responsabile	23
4.1.2.2	Amministratore	24
4.1.2.3	Analista	24
4.1.2.4	Progettista	24
4.1.2.5	Programmatore	24
4.1.2.6	Verificatore	25
4.1.3	Gestione delle comunicazioni	25

4.1.3.1	Comunicazioni interne	25
4.1.3.2	Comunicazioni esterne	25
4.1.4	Gestione degli incontri	25
4.1.4.1	Incontri interni	25
4.1.4.2	Incontri esterni	26
4.1.5	Versionamento	26
4.1.6	GitHub Workflow	26
4.1.7	Issues tracking	29

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

- https://it.wikipedia.org/wiki/ISO/IEC_12207
- 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.

Successivamente alla presentazione dei capitolati il gruppo *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 candidatura. Da queste discussioni emerge una preferenza per il progetto proposto dal referente Dr. Gregorio Piccoli. A seguito di questo viene organizzata una riunione con il proponente con l'obiettivo di approfondire e consolidare le richieste del capitolato.

In data 28 ottobre 2022 il gruppo *Catch em All* si candida a prendere in carico il progetto attraverso la lettera di candidatura.

Viene infine confermata l'assegnazione dell'appalto da parte del committente in data 04 novembre 2022.

2.2 Fornitura

2.2.1 Scopo

Il processo di fornitura richiederà al gruppo di definire le norme che dovranno essere rispettate per poter diventare un adeguato fornitore dell'azienda proponente *Zucchetti S.p.A.* e dei committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin. Di conseguenza verranno illustrati i prodotti e documenti che dovranno essere forniti per rispettare i requisiti concordati.

2.2.2 Rapporto con il proponente

Durante il corso del progetto il gruppo ha intenzione di instaurare un rapporto di collaborazione con il proponente Dr. Gregorio Piccoli in modo di:

- Approfondire gli aspetti chiave del progetto per far fronte ai suoi bisogni;
- Chiarire i vari dubbi che emergeranno durante il progetto;
- Definizione dei requisiti e vincoli da rispettare;
- Definire una stima dei costi, di tempo e denaro per la durata del progetto;
- Garantire che il prodotto soddisfi le richieste, accordandosi sulla qualifica di questo.

2.2.3 Prodotti e documenti forniti

2.2.3.1 Documenti per la candidatura

Al fine di scegliere il capitolato per il quale proporre la candidatura il gruppo ha organizzato degli incontri per la valutazione dei capitolati. Dopo aver scelto il capitolato i vari membri hanno stilato i vari documenti per la candidatura:

- **Lettera di candidatura:** Contiene l'impegno di svolgere il capitolato scelto e un riassunto dei preventivi contenuti nel documento *Preventivo_Costi_Ore_Rischi*;
- **Motivazione dei capitolati:** Contiene l'analisi fatta dai membri del gruppo sui 7 capitolati proposti, valutandone per ognuno i pro e contro e dando le motivazioni per il quale siano stati presi in considerazione o meno;
- **Preventivo costi ore e rischi:** Contiene un primo preventivo dei costi di tempo e denaro e dei rischi che si potrebbero incontrare durante il corso del progetto.

2.2.3.2 Analisi_dei_requisiti v 1.0.0

Questo documento stilato dagli analisti del gruppo contiene tutti i requisiti e casi d'uso individuati per il progetto. I seguenti sono ottenuti dal documento di presentazione del capitolato e in seguito integrati sia attraverso discussioni tra i membri del gruppo, sia organizzando incontri con il proponente.

2.2.3.3 Piano_di_progetto v 1.0.0

Descrizione

Questo documento stilato dal responsabile di progetto servirà ad organizzare le varie fasi del progetto individuate, di fare preventivi temporali e di costi su di esse e di compiere un'analisi dei rischi che si possono incontrare durante il corso del progetto.

Analisi dei rischi

In questa sezione si analizzano i vari tipi di rischio in cui si può incorrere durante la durata del progetto. Ogni rischio appartiene ad una specifica categoria, le quali sono:

- Rischi personali;
- Rischi tecnologico;
- Rischi organizzativi.

Ogni rischio è inoltre composto da:

- Nome;
- Descrizione;
- Identificazione;
- Precauzioni;
- Pericolosità;
- Stima di manifestazione;
- Conseguenze;
- Piano di contingenza.

Modello di sviluppo

In questa sezione viene specificato il modello di sviluppo che il team ha deciso di adottare, in questo caso il *modello incrementale*.

Pianificazione

In questa sezione sono contenute le pianificazioni temporali delle fasi in cui il responsabile di progetto ha deciso di suddividere quest'ultimo. Ogni fase è contraddistinta da:

- Nome identificativo;
- Descrizione;
- Periodo;
- Precondizioni;
- Postcondizioni;
- Attività;
- Ruoli attivi.

Ogni fase è inoltre suddivisa in vari periodi temporali per raggruppare al meglio varie attività che la compongono.

Infine ogni fase possiede un proprio diagramma di Gantt.

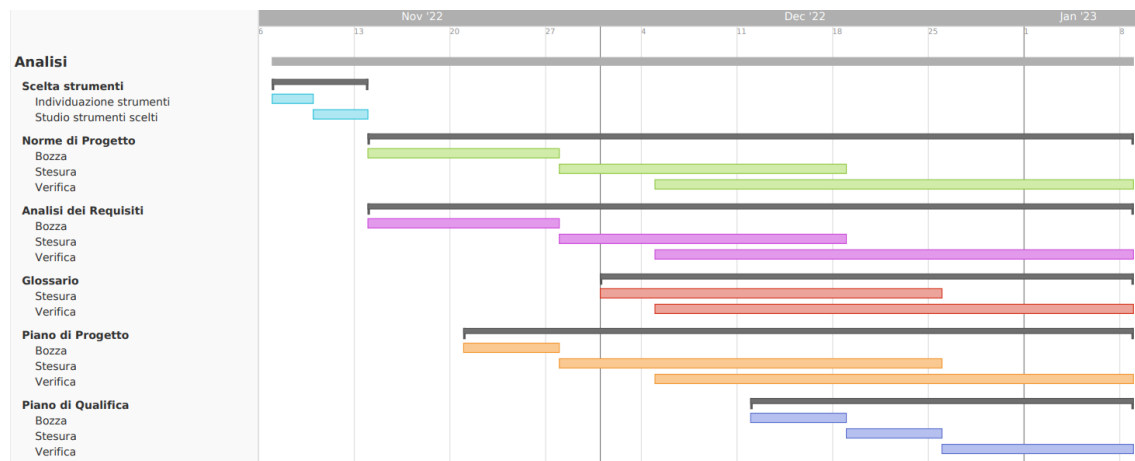


Figura 1: Esempio di diagramma di Gantt

Preventivi

In questa sezione sono contenuti i preventivi del numero di ore di lavoro che i vari periodi delle fasi identificate nella sezione di pianificazione richiedono. Inoltre si preventivano i vari costi delle fasi e il costo totale di progetto. Ogni preventivo sarà composto da:

- Due tabelle che indicano le ore e i costi necessarie per lo svolgimento del periodo;
- Un istogramma che illustra come sono state distribuite le ore fra i vari membri del gruppo;
- Un grafico a torta che mostra quanto ogni ruolo abbia inciso nel determinato periodo.

Consuntivi

In questa sezione vengono osservati il numero di ore di lavoro e costi reali di ogni periodo. Questi vengono poi relazionati con i preventivi fatti nella sezione *preventivi*.

2.2.3.4 Piano_di_qualifica v 1.0.0

Descrizione

Questo documento stilato dai membri con il ruolo di analista e verificatore contiene i vari obiettivi e metriche che devono permettere ai processi e prodotti del progetto essere verificati e validati, e di conseguenza garantirne la qualità.

Struttura documento

Questo documento è suddiviso in:

- **Obiettivi e metriche di qualità di:**
 - **Processo:** contiene i vari obiettivi generici e specifici e le metriche correlate ad essi che permettono la qualità di un processo;
 - **Prodotto:** contiene i vari obiettivi e le metriche correlate ad essi che permettono la qualità di un prodotto;
- **Specifiche dei test:** vengono definiti i vari test che dovranno essere eseguiti i quali sono:
 - Test di unità;
 - Test di integrazione;
 - Test di sistema;
 - Test di regressione;
 - Test di collaudo.
- **Risultati dei test:** dove vengono illustrati i risultati dei test definiti nella sezione *Specifiche dei test*.

Struttura obiettivi

Ogni obiettivo sarà contrassegnato da un codice univoco così composto:

OQ<<Tipo di obiettivo>><<Genericità>><<ID>>

Dove

- OQ sta per obiettivo di qualità;
- <<Tipo di obiettivo>> identifica se è di processo o prodotto (PC-PD);

- <<Genericità>> è esclusivo per gli obiettivi di processo e indica se un obiettivo è da applicare ad ogni processo o se ne riguarda uno specifico (G-S);
- <<ID>> è un contatore correlato al tipo di obiettivo.

Struttura metriche

Ogni metrica sarà contrassegnata da un codice univoco così composto:

MQ<<Tipo di metrica>><<Genericità>><<ID>>

Dove

- MQ sta per metrica di qualità;
- <<Tipo di metrica>> identifica se è di processo o prodotto (PC-PD);
- <<Genericità>> è esclusivo per le metriche di processo e indica se una metrica è da applicare ad ogni processo o se ne riguarda uno specifico (G-S);
- <<ID>> è un contatore correlato al tipo di metrica.

2.2.3.5 Proof of Concept

Un software esempio che va ad analizzare alcune sezioni critiche per lo sviluppo del progetto, osservate in seguito ad un'analisi del gruppo. Questo software permetterà al gruppo di determinare la fattibilità pratica e l'applicabilità di alcuni concetti necessari per la progettazione e codifica del prodotto finale.

2.3 Sviluppo

2.3.1 Scopo

L'obiettivo del processo di sviluppo è definire le attività che il gruppo deve eseguire per realizzare il prodotto finale richiesto dal proponente.

2.3.2 Analisi dei requisiti

2.3.2.1 Scopo e descrizione

In questa fase vengono stilati tutti i requisiti che saranno necessari per la successiva fase di progettazione e quindi per lo sviluppo di un prodotto che risponda in maniera completa ai bisogni del proponente. Il documento stilato in questa fase contiene:

- Una descrizione generale del prodotto;
- L'analisi dettagliata dei casi d'uso;
- I requisiti individuati tramite:
 - Documento di presentazione del capitolato;
 - Confronti tra i membri del gruppo;
 - Incontri con il proponente.

2.3.2.2 Struttura dei casi d'uso

Ogni caso d'uso è identificato utilizzando la seguente convenzione di nomenclatura:

UC<<ID_CasoBase>>.<<ID_SottoCaso>>

Dove

- <<ID>> identifica l'uso case;
- <<ID_SottoCaso>> identifica eventuali sottocasi.

Ogni caso d'uso è composto inoltre da:

- Descrizione: una breve descrizione dell'attività rappresentata dal caso d'uso;
- Attori: entità esterne al sistema che interagiscono con esso. Ne esistono di due tipologie:
 - Primario: interagisce con il sistema per raggiungere un obiettivo;
 - Secondario: aiuta l'attore primario a raggiungere l'obiettivo.
- Precondizione: descrive lo stato del sistema prima dell'attività svolta nel caso d'uso;
- Postcondizione: descrive lo stato del sistema dopo l'attività svolta nel caso d'uso;
- Scenario principale: elenco che descrive il flusso degli eventi dell'attività rappresentata dal caso d'uso;
- Scenari alternativi (se presenti): elenco che descrive gli eventi del caso d'uso dopo un imprevisto che lo ha deviato dallo scenario principale;
- Scenari inclusi (se presenti): elenco di casi d'uso che svolgono attività necessarie allo svolgimento dello scenario principale;
- Generalizzazioni (se presenti): elenco di casi d'uso che generalizzano il caso d'uso principale.

2.3.2.3 Struttura dei requisiti

Ogni requisito è identificato da un codice univoco così composto:

R<<TIPOLOGIA DI REQUISITO>>-<<ID>>

Dove:

- <<TIPOLOGIA DI REQUISITO>> identifica una classe tra le seguenti:
 - Funzionale {F};
 - Qualità {Q};
 - Vincolo {V};
 - Prestazionale {P}.
- <<ID>> identifica numericamente il requisito nella classe di appartenenza

Nel documento vengono raggruppati per categoria specificandone:

- Grado di obbligatorietà;
- Descrizione;
- Fonti, le quali possono essere:
 - Il capitolato d'appalto;
 - Verbali interni;
 - Verbali esterni;
 - I casi d'uso identificati.

2.3.3 Progettazione

2.3.3.1 Scopo e descrizione

Lo scopo di questa fase è quello di individuare le varie caratteristiche che comporranno il prodotto richiesto dal proponente. Queste verranno individuate attraverso i vari requisiti e casi d'uso identificati nella fase *Analisi dei requisiti*. Le varie caratteristiche verranno poi messe insieme per costruire una singola soluzione che rispetti i vari obiettivi di qualità del prodotto.

2.3.3.2 Suddivisione attività

- **Proof of concept;**
- **Progettazione architettuale;**
- **Progettazione di dettaglio.**

2.3.3.3 Proof of concept

Scopo

In questa fase viene prodotto un software esempio che sarà anche la technology baseline del prodotto finale. Questo andrà ad analizzare alcune sezioni critiche per lo sviluppo del progetto e servirà ad agevolare le successive scelte di progettazione del gruppo, aiutando a determinare la fattibilità e l'applicabilità di alcune scelte analizzate.

Suddivisione periodi

Questa fase è divisa in due periodi:

- Periodo nel quale vengono identificati i requisiti del POC e delle tecnologie necessarie a svilupparlo, oltre che lo studio di queste ultime;
- Periodo di produzione del POC.

2.3.3.4 Progettazione architetturale

Scopo

Lo scopo di questa fase è il raffinamento della technology baseline definita nella fase di *Proof of Concept*, e discute ad alto livello l'architettura del prodotto e delle sue componenti. Le scelte che il gruppo effettua in questa fase riguarderanno la struttura complessiva del sistema e ne influenzeranno varie caratteristiche qualitative come per esempio l'efficienza, l'estensibilità e la manutenibilità.

2.3.3.5 Progettazione di dettaglio

Scopo

Lo scopo di questa fase è definire le specifiche di dettaglio dell'architettura del prodotto e di tutte le sue componenti, scomposte in unità. Queste saranno correlate a diagrammi UML che le descriveranno e ai test di verifica per la qualità, i quali saranno indicati nel documento *Piano di qualifica v1.0.0*. Tali informazioni costituiranno la Product Baseline, la quale conterrà:

- Desing patterns utilizzati;
- Definizione delle classi;
- Diagrammi UML:
 - Diagrammi delle attività;
 - Diagrammi delle classi;
 - Diagrammi di sequenza.
- Test di verifica per ogni componente.

2.3.4 Codifica

2.3.4.1 Scopo e descrizione

La fase di *Codifica* è assegnata ai membri con il ruolo di programmatore, i quali dovranno realizzare il prodotto software richiesto dal proponente utilizzando ciò che i progettisti hanno definito nella fase di *Progettazione*.

Per garantire un prodotto adeguato il codice dovrà essere verificato in modo che rispetti le metriche che garantiscono gli obiettivi di qualità definiti nel documento *Piano di qualifica v 1.0.0*.

2.3.4.2 Suddivisione attività

Le varie attività che comporranno la fase di codifica saranno aggiunte più avanti.

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo di questa sezione è definire gli standard necessari 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 svolta da almeno una persona. Il documento è considerato verificato quando i Verificatori dichiarano che le modifiche necessarie 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;

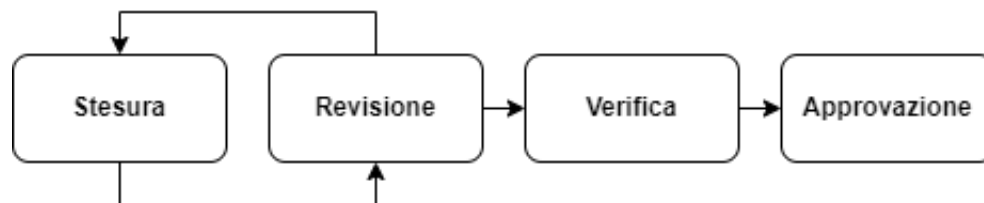


Figura 2: Ciclo di vita dei documenti

3.1.3 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.

3.1.3.1 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 nomenclatura 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 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.

3.1.3.3 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 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 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 *Verbalì interni*;
- **Uso esterno:** sono documenti di interesse a tutti gli stakeholder, tra cui *Analisi dei requisiti*, *Verbalì 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 eccezione 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:

- **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;
- **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 data 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 scelta 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 estensione diversa da quelle citate sono escluse attraverso il file **.gitignore**.

3.3 Assicurazione della qualità

3.3.1 Scopo

Questo processo ha lo scopo di assicurare che tutti i processi e prodotti siano conformi con gli obiettivi e le metriche definiti dal gruppo nel documento *Piano_di_qualifica v 1.0.0*. Devono essere continuamente osservate la qualità di processo per garantire una buona gestione del progetto e la qualità del prodotto per assicurarsi di lavorare in modo conforme alle richieste del proponente.

3.3.2 Denominazione obiettivi di qualità

Ogni obiettivo sarà contrassegnato da un codice univoco così composto:

OQ<<Tipo di obiettivo>><<Genericità>><<ID>>

Dove

- OQ sta per obiettivo di qualità;
- <<Tipo di obiettivo>> identifica se è di processo o prodotto (PC-PD);
- <<Genericità>> è esclusivo per gli obiettivi di processo e indica se un obiettivo è da applicare ad ogni processo o se ne riguarda uno specifico (G-S);
- <<ID>> è un contatore correlato al tipo di obiettivo.

3.3.3 Denominazione metriche di qualità

Ogni metrica sarà contrassegnata da un codice univoco così composto:

MQ<<Tipo di metrica>><<Genericità>><<ID>>

Dove

- MQ sta per metrica di qualità;
- <<Tipo di metrica>> identifica se è di processo o prodotto (PC-PD);
- <<Genericità>> è esclusivo per le metriche di processo e indica se una metrica è da applicare ad ogni processo o se ne riguarda uno specifico (G-S);
- <<ID>> è un contatore correlato al tipo di metrica.

3.4 Verifica

3.4.1 Scopo

Questo processo ha lo scopo di confermare che ciascuna attività svolta soddisfi i requisiti e gli obiettivi specificati, attraverso le metriche scelte dal gruppo e descritte in dettaglio nel documento *Piano_di_qualifica v 1.0.0* e che non abbia introdotto errori. Il processo di *Verifica* deve essere integrato nei processi di *Fornitura*, *Sviluppo* e *Manutenzione*.

Il processo di verifica è suddiviso in due fasi:

- **Analisi statica**, la quale non richiede l'esecuzione dell'oggetto di verifica;
- **Analisi dinamica**, la quale richiede l'esecuzione dell'oggetto di verifica.

3.4.2 Analisi statica

L'analisi statica si occupa di analizzare documentazione e codice e accerta conformità a regole, assenza di errori, completezza dei requisiti desiderati. Inoltre poiché non richiede l'esecuzione dell'oggetto di verifica, si può applicare ad ogni prodotto di processo.

Si utilizzano due metodi per svolgere analisi statica:

- **Walkthrough**;
- **Inspection**.

Queste sono effettuate tramite studio dell'oggetto di verifica e lettura umana o automatizzata.

3.4.2.1 Walkthrough

Scopo

I verificatori, insieme agli sviluppatori quando necessario, che utilizzano questo metodo devono rilevare la presenza di errori attraverso una lettura critica ad ampio spettro del prodotto da analizzare. Questo metodo è molto oneroso dal punto di vista delle risorse utilizzate, e perciò si cercherà di utilizzare solo fino al momento in cui non sarà disponibile una checklist.

Fasi

1. Pianificazione, svolta da autori e verificatori;
2. Lettura, svolta dai verificatori;
3. Discussione, svolta da autori e verificatori;
4. Correzione degli errori, svolta dagli autori.

3.4.2.2 Inspection

Scopo

I verificatori che utilizzano questo metodo devono rilevare la presenza di errori eseguendo una lettura mirata dell'oggetto di verifica attraverso l'utilizzo di una checklist. Si cerca quindi di immaginare in precedenza quali saranno le criticità dell'oggetto da analizzare e di elencarli.

Fasi

1. Pianificazione;
2. Definizione di una checklist;
3. Lettura;
4. Correzione degli errori, svolta dagli autori.

3.4.3 Analisi dinamica

L'analisi dinamica si occupa di eseguire dei test sugli oggetti di verifica che devono essere eseguiti. Questo permetterà al gruppo di accertarsi dell'assenza di errori. I test dovranno essere:

- **Ripetibili**, ovvero che garantiscano la correttezza dell'oggetto di verifica e quindi la rimozione di eventuali errori.
- **Automatizzati**, ovvero svolti in maniera automatica da precisi strumenti selezionati.

Inoltre verranno eseguiti diversi tipi di test:

- **Test di unità**;
- **Test di integrazione**;
- **Test di sistema**;
- **Test di regressione**;
- **Test di accettazione e collaudo**.

3.4.4 Denominazione test di verifica

Ogni test sarà contrassegnato da un codice univoco così composto:

TV<<Tipo di test>><<ID>>

Dove

- TV sta per test di verifica;
- <<Tipo di test>> identifica il tipo di test che si vuole fare, questi possono essere:
 - UN, di unità;

- IN, di integrazione;
 - ST, di sistema;
 - RG, di regressione;
 - AC, di accettazione e collaudo.
- <<ID>> è un contatore correlato al tipo di test.

Informazioni più dettagliate sui vari test, strumenti e metriche utilizzate per la verifica si possono trovare nel documento *Piano_di_qualifica v 1.0.0*.

3.5 Validazione e collaudo

3.5.1 Scopo

Lo scopo di questo processo è quello di confermare che tutti i requisiti siano rispettati all'interno del prodotto finale, dopo aver svolto tutti i test di verifica. Questi due processi sono strettamente legati fra loro e una buona verifica durante il corso del progetto permette di superare anche il processo di validazione. Questo processo permette anche di confermare che il prodotto finale sia conforme alle richieste del proponente.

3.6 Usabilità

3.6.1 Scopo

Questo processo ha lo scopo di assicurare che siano prese in considerazione, ed opportunamente indirizzate, le considerazioni espresse dalle parti interessate, gli stakeholders, relativamente alla facilità d'uso del prodotto finale da parte degli utenti cui è rivolto, al supporto che ne riceverà, alla formazione, all'incremento della produttività, alla qualità del lavoro, all'accettazione del prodotto stesso.

4 Processi organizzativi

4.1 Gestione progetto

4.1.1 Scopo

Lo scopo di questo processo è quello di fornire al gruppo delle linee guida su come gestire l'organizzazione delle varie fasi del progetto, con i punti a seguito elencati:

- Gestione dei ruoli;
- Gestione delle comunicazioni;
- Gestione degli incontri;
- Gestione per il controllo della versione;
- Gestione del GitHub Workflow;
- Gestione del tracciamento delle attività.

4.1.2 Gestione ruoli

Durante il corso del progetto il gruppo dovrà gestire i vari ruoli garantendo una suddivisione equa tra i membri e pertinente rispetto al processo in svolgimento.

I sei ruoli possibili da assegnare sono:

- Responsabile;
- Amministratore;
- Analista;
- Progettista;
- Programmatore;
- Verificatore.

4.1.2.1 Responsabile

Il responsabile di progetto è la figura di riferimento con l'esterno, quindi garantisce una buona comunicazione con proponente e committente. Si assume inoltre la responsabilità delle scelte del gruppo, dopo averle approvate.

Il responsabile quindi gestisce:

- Elaborazione di piani e scadenze del progetto;
- Gestire la suddivisione dei ruoli all'interno del gruppo;
- Approvare il rilascio di prodotti parziali o finali, come documentazione o software;
- Analisi e gestione dei rischi.

4.1.2.2 Amministratore

L'amministratore di progetto ha lo scopo di controllare l'efficienza dell'ambiente di lavoro, assicurandosi quindi che gli strumenti di supporto alle norme di progetto siano usate correttamente dal gruppo.

L'amministratore quindi gestisce:

- La corretta applicazione delle norme di progetto;
- La ricerca di nuovi metodi per rendere più efficiente l'ambiente di lavoro;
- Le versioni dei vari prodotti durante il corso del progetto;
- L'analisi di metodi per la gestione della qualità.

4.1.2.3 Analista

L'analista si occupa di analizzare a fondo il problema e di individuare i vari requisiti che dovrà avere il prodotto finale in base a ciò che si ricava dal capitolato e dai successivi incontri con il proponente.

L'analista quindi gestisce:

- Lo studio approfondito del dominio del problema;
- Il documento *Analisi dei requisiti*;
- L'individuazione dei requisiti del prodotto finale;
- L'analisi dei casi d'uso.

4.1.2.4 Progettista

Il progettista si occupa di trovare soluzioni tecniche e tecnologiche che possano permettere al gruppo di creare un prodotto che rispetti al meglio tutti i requisiti individuati dagli analisti.

Il progettista quindi gestisce:

- La scelta degli aspetti tecnici e tecnologici per la realizzazione del prodotto;
- La scelta dei vari modelli da utilizzare nella definizione dell'architettura del prodotto;
- Di definire l'architettura del prodotto che verrà poi programmato;

4.1.2.5 Programmatore

Il programmatore si occupa di codificare le soluzioni individuate dai progettisti per la creazione del prodotto finale.

Il programmatore quindi gestisce:

- La scrittura del codice in modo che sia chiaro e facile da mantenere;
- Gli strumenti che si occupano dei test utilizzati per la verifica e la validazione del software;
- La redazione del Manuale Utente relativo alla codifica del prodotto;

4.1.2.6 Verificatore

Il verificatore si occupa di tutte le operazioni di verifica e validazione dei vari prodotti parziali e finali del progetto.

Il verificatore quindi gestisce:

- La verifica e validazione dei prodotti parziali o finali in fase di revisione in modo che rispettino gli obiettivi di qualità imposti nel documento *Piano_di_qualifica v 1.0.0*. I seguenti prodotti se a norma verranno di conseguenza integrati;
- Segnalare gli eventuali errori riscontrati.

4.1.3 Gestione delle comunicazioni

4.1.3.1 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*

4.1.3.2 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

4.1.4 Gestione degli incontri

4.1.4.1 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 schedare 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.

4.1.4.2 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.

4.1.5 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).

4.1.6 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`

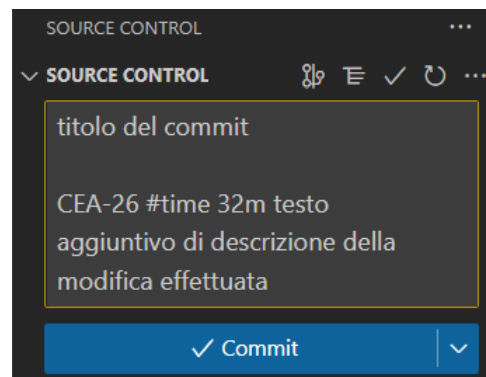
Questo permette di identificare titolo e numero della issue di appartenenza. Una volta fatto ciò viene creato un **primo commit**, nel cui messaggio è specificata l'avvenuta presa in carico della issue, la quale dovrà passare dallo stato "to do" allo stato "in progress"

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):

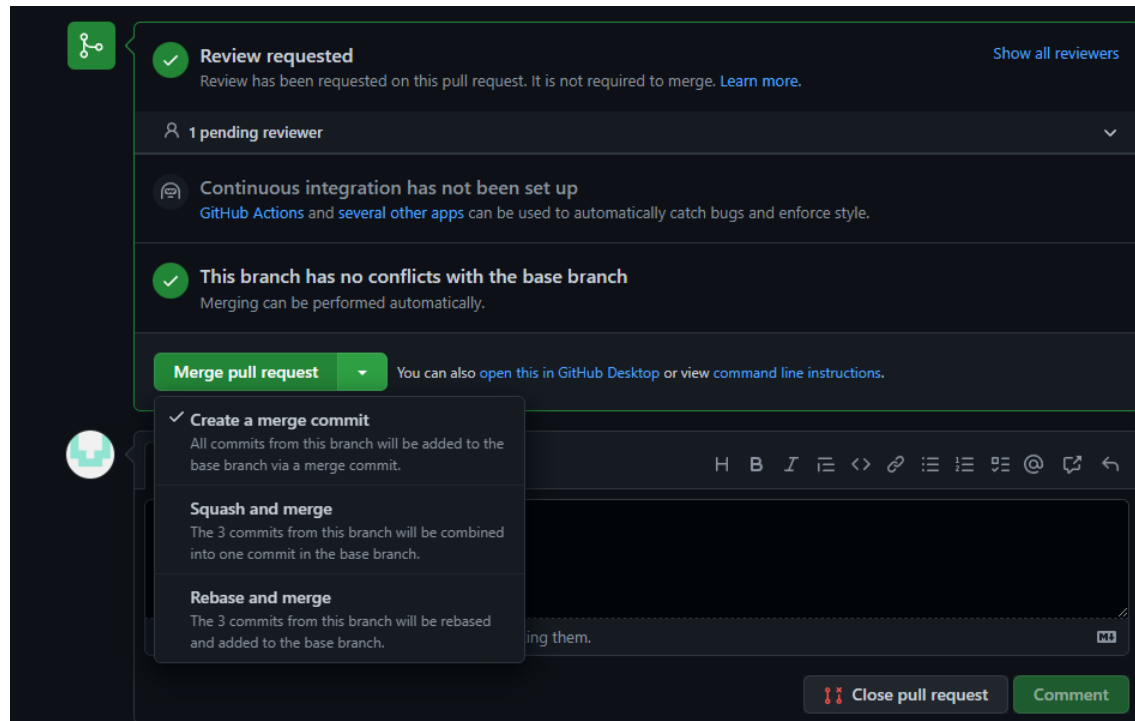
```
CEA-num #review #time ww dd hh mm <testo aggiuntivo>
```

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

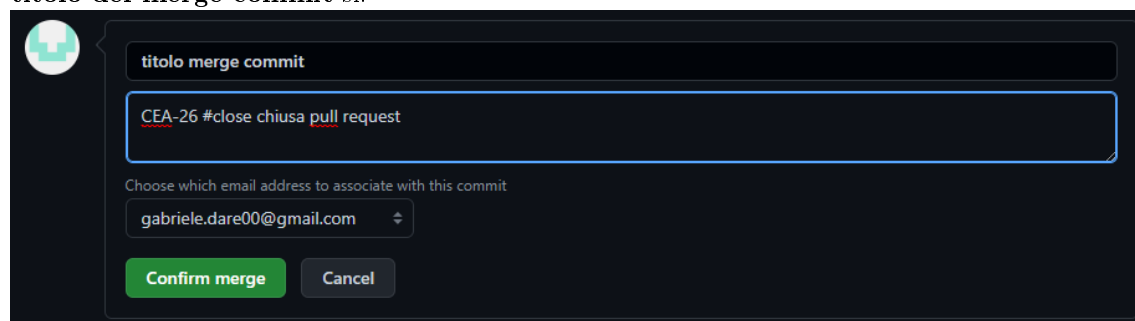
Per permettere la revisione è necessario aprire una pull request, il titolo deve corrispondere al nome del branch. Una volta revisionata la issue, se presenta qualche problema può essere spostata allo stato "in progress" dal pannello JIRA. Altrimenti attraverso una pull request nel ramo "main" e con 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.



4.1.7 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"