Università degli Studi di Padova

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

Norme di progetto

Versione	(0.1.2)
Approvazione	(Da dare)
Redazione	(Gabriele Da Re, Zhen Wei Zheng e Luca Brugnera)
Verifica	(Sinicato Nicola, Gabriele Da Re)
Stato	(Da approvare)
Uso	(modifica)
Distribuzione	(modifica)

Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.1.2	11/01/2023	Aggiunto punto nella verifica statica	Gabriele Da Re	Verificatore
0.1.1	11/01/2023	Fine della verifica globale del documento	Sinicato Nicola, Gabriele Da Re	Verificatore, Verifcatore
0.1.0	10/01/2023	Inizio della verifica globale del documento	Sinicato Nicola	Verificatore
0.0.9	10/01/2023	Aggiunta sezione "File correctness script_{G}"	Gabriele Da Re	Amministratore
0.0.8	02/01/2023	Aggiunte appendici	Matteo Stocco	Amministratore
0.0.7	28/12/2022	Aggiunte parti nella sezione "Processi di supporto" e "Processi organizzativi"	Matteo Stocco	Amministratore
0.0.6	24/12/2022	Completato sezione "Processi primari"	Matteo Stocco	Amministratore
0.0.5	29/11/2022	Completato sezione "Documentazione" e revisione	Zhen Wei Zheng, Nicola Sinicato	Amministratore, Verificatore
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

Elenco delle figure

1 Introduzione

1.1 Scopo del documento

Questo documento ha come obiettivo il fissare gli standard che permetteranno al gruppo *Catch Em All* di garantire qualità al prodotto$_G$ e ai processi durante l'intera durata del progetto. Verranno quindi definiti metodi di verifica e validazione$_G$ continui che permetteranno al gruppo di agire in modo rapido e incisivo nel momento in cui si dovranno fare delle correzioni su eventuali errori o andamenti indesiderati. Questo allo scopo di sprecare meno risorse possibili e produrre un prodotto che sia facilmente mantenibile.

1.2 Scopo del prodotto

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

- Registrazione presso siti web;
- Creazione di spam$_G$;
- Violare sistemi di sicurezza.

I bot$_G$, 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$_G$, 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$_G$, 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$_G$ proposte, che portano alla creazione di enormi dataset$_G$ 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 un sistema CAPTCHA che sia in grado di distinguere un utente umano da un bot$_G$.

1.3 Glossario

Per evitare ambiguità relative al linguaggio utilizzato nei documenti prodotti, viene fornito il documento **Glossario v1.0.0**. Qui vi sono contenuti tutti i termini specifici al dominio del problema, i quali avranno una definizione che servirà per comprenderne al meglio il loro significato. Ogni termine che avrà un riferimento al glossario dovrà avere una *G* come pedice.

1.4 Riferimenti

1.4.1 Riferimenti normativi:

- Capitolato C1 "CAPTCHA: umano o sovrumano?" `https://www.math.unipd.it/~tullio/IS-1/2022/Progetto/C1.pdf`

1.4.2 Riferimenti informativi:

- Processi di ciclo di vita - Materiale didattico del corso di Ingegneria del Software: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T02.pdf`;
- Il ciclo di vita del Software - Materiale didattico del corso di Ingegneria del Software: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T03.pdf`;
- Gestione di progetto - Materiale didattico del corso di Ingegneria del Software: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T04.pdf`;
- `https://it.wikipedia.org/wiki/ISO/IEC_12207`;
- Approfondimento standard ISO/IEC 12207: `https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf`;
- Qualità di prodotto - Materiale didattico del corso di Ingegneria del Software: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T08.pdf`;
- Qualità di processo - Materiale didattico del corso di Ingegneria del Software: `https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T09.pdf`;
- Standard SQuaRE: `http://www.iso25000.it/styled/`;
- Standard SPICE: `https://it.frwiki.wiki/wiki/ISO/CEI_15504`;
- 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 primo preventivo *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 le quali siano stati presi in considerazione o meno;
- **Preventivo costi ore e rischi**: Contiene un primo preventivo dei costi di tempo e il relativo costo in denaro e una prima bozza dell'analisi 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 diversi rischi in cui il team può incombere durante tutta la durata del progetto. Ogni rischio appartiene ad una specifica categoria, le quali sono:

- Rischi personali;
- Rischi tecnologici;
- 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 le diverse attività che la compongono.
Infine ogni fase possiede un proprio diagramma di Gantt.

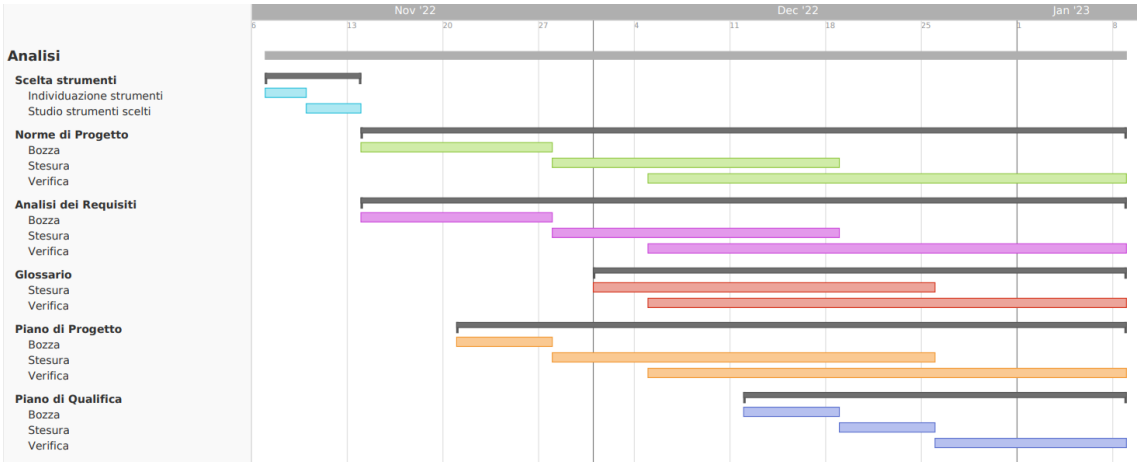

Figura 1: Esempio di diagramma di Gantt

Preventivo
In questa sezione sono contenuti i preventivi sulla distribuzione oraria del lavoro assegnato ad ogni periodo di ogni fase. Inoltre viene preventivato il costo di ogni fase e il costo totale del 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.

Consuntivo
Iin questa sezione vengono indicati il numero di ore di lavoro impiegate e i relativi costi effettivi di ogni periodo. Questi vengono poi relazionati con i preventivi fatti nella sezione *preventivo*.

2.2.3.4 Piano_di_qualifica v 1.0.0

Descrizione
Questo documento stilato dai membri con il ruolo di analista e di verificatore contiene i vari obiettivi e metriche che permettono di garantire la qualità della verifica e della validazione dei processi e dei prodotti del progetto.

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>><<ID>>`

Dove:

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

Dove:

- `MQ` sta per metrica di qualità;
- `<<Tipo di metrica>>` identifica se è di processo o prodotto (PC-PD);
- `<<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'use 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$_G$: 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 architetturale**;
- **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 il seguente ciclo di vita:

- **Stesura**: Il documento viene scritto utilizzando la metodologia AGILE, adottando sprint_{G} di durata settimanale;
- **Revisione**: Alla fine di ogni sprint_{G} i documenti modificati devono essere revisionati da un membro del gruppo diverso dal redattore. Solo dopo una revisione con esito positivo le modifiche e i nuovi contenuti possono essere integrati nel documento finale;
- **Verifica**: Avviene in un periodo della fase di analisi indicata nel documento *Piano_di_progetto v 1.0.0*. Tale attività viene svolta da almeno una persona. Il documento è considerato verificato quando i Verificatori dichiarano che le modifiche necessarie per renderlo coerente con tutte le norme sono state 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.

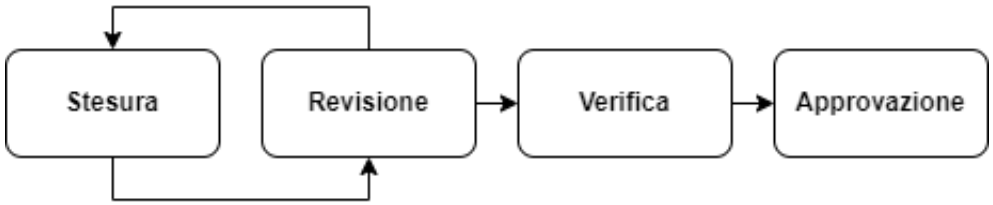

Figura 2: Ciclo di vita dei documenti

3.1.3 Struttura dei documenti

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

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

3.1.3.1 Frontespizio

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

- **Logo dell'università**: Logo dell'*Università di Padova* posizionato nella parte centrale alta della pagina, seguito dalla nomenclatura "Università degli Studi di Padova";
- **Logo del gruppo**: Logo del gruppo, posizionato in centro, a seguito della nomenclatura dell'università;
- **Nome del gruppo e del progetto**: Il nome del gruppo e del capitolato scelto, seguito dal recapito email del gruppo;
- **Nome del documento**: Il titolo del documento, definito in grassetto e posizionato al centro della pagina;
- **Tabella di descrizione**: 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. Questo è impostato come segue:

- **Versione**: Indica la versione del documento dopo una particolare modifica;
- **Descrizione**: Descrive brevemente la modifica apportata;
- **Data**: Indica la data in cui è stato modificato il documento.

3.1.3.3 Indice

Per agevolare la lettura, tutti i documenti sono dotati di un indice. Le sezioni sono rappresentate da un numero identificativo seguito dal titolo della sezione. Ogni sottosezione deve riportare il numero della sezione genitore e poi il proprio numero identificativo. I numeri partono dall'uno.

3.1.3.4 Contenuto principale

Le varie pagine di contenuto sono costituite da:

- **Intestazione**: In alto a sinistra deve esserci il nome del gruppo *Catch em All*, mentre in altro a destra si trova il numero e nome della sezione in cui ci si trova;
- **Pie di pagina**: In basso sinistra si trova il nome del documento e la sua versione attuale, mentre in basso a destra viene indicato il numero della pagina in cui ci si trova insieme al 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, questi sono *Norme di progetto* e *Verbali interni*;
- **Uso esterno**: Sono documenti di interesse a tutti gli stakeholder, questi sono *Analisi dei requisiti*, *Verbali esterni*, *Piano di progetto*, *Piano di qualifica*, *Glossario*.

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 maiuscola;
- Se il nome comprende più parole allora ognuna di esse è separata dal simbolo '_';
- Deve essere seguito dalla versione in cui si trova.

La sigla della versione deve essere così strutturata:

v.X.Y.Z

Dove:

- **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 dei verificatori, 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**.

Esempio corretto: Norme_di_progetto_v.0.0.1, Norme_di_progetto_v.0.2.1, Norme_di_progetto_v.1.0.0.
Esempi non corretti: Norme_di_progetto, NormeDiProgetto.
I verbali non seguono questa norma e hanno una nomenclatura diversa, poiché non subiscono variazioni dopo la prima redazione e hanno il seguente formato:

`V<<Tipo di verbale>>_<<Data verbale>>`

Dove:

- `V` sta a indicare che si tratta di un verbale;
- `Tipo di verbale` indica se è interno o esterno;
- `Data verbale` indica la data in cui è stato redatto, ed deve essere scritta nel formato YYYYMMDD.

3.1.5.2 Stile di testo

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

- **Grassetto**: Utilizzato per i termini da descrivere all'interno degli elenchi puntati e per i titoli delle varie sezioni del documento. Può essere utilizzato anche per evidenziare concetti particolarmente rilevanti;
- **Corsivo**: Utilizzato per il nome e l'email del gruppo, il nome del progetto, riferimenti ad altri documenti e sigle;
- **Link**: Sono collegamenti esterni al documento.

3.1.5.3 Glossario

Le norme da seguire relative al *Glossario* sono:

- Ogni parola presente nel documento *Glossario v 1.0.0* viene contrassegnata con una 'G' a pedice all'interno dei vari documenti dove è utilizzata;
- 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 utilizzare 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, la quale deve iniziare con la lettera maiuscola.

3.1.5.5 Sigle

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*;
  - **Gls**: Indica il *Glossario*.
- **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*.
- **Revisioni di progetto**:
  - **RTB**: Indica la prima revisione di progetto, e comprende la Requirements and Technology Baseline;
  - **PB**: Indica la seconda revisione, e comprende la Product Baseline;
  - **CA**: Indica la terza revisione, e comprende la Customer Acceptance.

3.1.5.6 Formato della data

Il team ha adottato il seguente formato per le date all'interno dei documenti:

DD-MM-YYYY

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

3.1.6 Elementi grafici

3.1.6.1 Tabelle

Ogni tabella del documento deve:

- Essere centrata orizzontalmente;
- Essere accompagnata da una didascalia che indichi il numero della tabella all'interno del documento. L'unica eccezione è la tabella delle modifiche che non necessita di didascalie.

3.1.6.2 Immagini

Ogni immagine del documento deve:

- Essere centrata orizzontalmente;
- Essere accompagnata da una didascalia che indichi il numero dell'immagine all'interno del documento.

3.1.7 Strumenti

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

- **LaTeX**: Per la produzione dei documenti, il team ha deciso di usare il linguaggio di markup *LaTeX*;
- **Microsoft Word**: Per la stesura delle bozze di alcune parti di documenti;
- **Microsoft Excel**: Per la creazione delle tabelle con i preventivi delle ore e costi che verranno poi inserite nel *PdP*;
- **StarUML**: Per la creazione dei vari UML da inserire all'interno dei documenti;
- **LucidChart**: Per discussioni di gruppo sui vari UML creati, dato che lo strumento permette modifiche condivise.

3.2 Gestione della configurazione

3.2.1 Scopo

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

3.2.2 Sistemi software utilizzati

Per la gestione dei file e gli aggiornamenti dei documenti viene utilizzato il sistema di controllo di versione Git, attraverso il servizio di GitHub_G. Per la scrittura dei verbali interni ed esterni si è deciso di utilizzare il servizio Confluence offerto da JIRA_G.

3.2.3 Struttura del repository

Di seguito viene data una lista di cartelle presenti nella repository su Git:

- **catchEmAll-SWE/catchEmAll-Docs** è la repository contenente documentazione riguardante:
  - **Assegnazione dell'appalto**: Con al suo interno i documenti *LetteraCandidatura*, *Preventivo_ Ore_ Costi_ rischi*, *motivazione_ capitolati*;
  - **Diari di bordo**;
  - **Norme di progetto**;
  - **Analisi dei requisiti**;
  - **Piano di progetto**;
  - **Piano di qualifica**;
  - **Glossario**;
  - **Link dei verbali interni ed esterni**.

3.2.4 Struttura delle cartelle su Confluence

Confluence, un servizio fornito da JIRA_G è utilizzato dal gruppo per la scrittura e l'organizzazione di:

- **Verbali**: Interni ed esterni;
- **Sprint_G retrospective**: Contenente le varie analisi retrospettive del gruppo sugli sprint_G svolti;
- **Ricerche e documentazioni**: Contiene le documentazioni e analisi che il gruppo ha svolto su strumenti o tecnologie che possono essere utili allo sviluppo del progetto.

Tale scelta è stata guidata dalla qualità dei template forniti dallo strumento, i quali facilitano la scrittura e la comprensione di alcuni specifici documenti.

3.2.5 Gestione delle modifiche

Per evitare i conflitti tra le modifiche, mantenere in ordine i file e garantire che all'interno del branch_G principale ci siano solo documenti verificati, il gruppo ha deciso che ogni qual volta sia necessaria una modifica in uno specifico documento deve essere creato un branch_G per apportarla. La modifica in questione potrà essere integrata nel branch_G principale solo dopo essere stata revisionata dal *Ve*.

3.2.6 Tipi di file presenti

Nella repository *catchEmAll-Docs* sono presenti esclusivamente file **.tex**, **.png** e **.pdf**. Altri file prodotti durante la stesura dei documenti con estensioni diverse 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$_G$ 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>><<ID>>`

Dove:

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

Dove:

- `MQ` sta per metrica di qualità;
- `<<Tipo di metrica>>` identifica se è di processo o prodotto (PC-PD);
- `<<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 la documentazione e il codice e accerta la conformità alle regole introdotte, l'assenza di errori, la 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.

File correctness script$_G$
Allo scopo di eseguire in modo automatizzato l'inspection dei file è stato scritto un programma in python che permette di verificare determinati elementi:

- All'interno del file;
- Nomi dei file;
- Estensioni dei file e numero;
- Nomi di cartelle.

Queste verifiche sono per la maggior parte delegate ad un modulo chiamato "specific_rules.py" il quale contiene tutte le verifiche da fare tramite espresssioni regolari (regex) su vari input forniti alle funzioni del file. Questo modulo viene incluso da un altro modulo "checks.py" il quale a sua volta è incluso dallo script$_G$ "file_correctness.py".
Le verifiche effettuate dallo script$_G$ sono:

1. Presenza di tutte le directory;
2. Correttezza nome delle directory;
3. Sottoalbero delle directory di documentazione principale presente e corretto;
4. Presenza di un solo file pdf e di nessun altro file;
5. Presenza di un solo file LaTeX nella cartella "src" e nessun altro file;
6. File LaTeX nella cartella "src" con determinati parametri;
   In particolare:

```
'font size': r'.*\\documentclass\[10pt\]\{article\}.*',
'packages file': r'.*\\input\{sections/packages\}.*',
'style file': r'.*\\input\{sections/style\}.*',
'title page': r'.*\\input\{sections/title_page\}.*',
'page roman numbering': r'.*\\pagenumbering\{roman\}.*',
'table of contents': r'.*\\tableofcontents.*',
```

7. Controllo della presenza di almeno un file LaTeX nella cartella "sections" e di nessun altro tipo di file;
8. Controllo della presenza dei file necessari nella cartella "sections", precedentemente inclusi nel file LaTeX in "src"; Ovvero:

```
necessary_sections_files = [
'style.tex',
'packages.tex',
'title_page.tex',
'modifiche.tex',
]
```

9. Controllo della presenza di determinati parametri nel file title_page.tex, tra cui il titolo che coincida con il nome della cartella in cui si trova il file;
10. Controllo della presenza di determinati parametri nel file style.tex, tra cui la correttezza del nome del file;
11. Controllo della presenza di determinati parametri nel file modifiche.tex, tra cui il giusto ordine e l'assenza di ripetizioni nel numero delle versioni;
12. Controllo della presenza dell'ultima versione nei file title_page.tex e in style.tex;
13. Controllo del nome e della correttezza della versione del file pdf;
14. Controllo del rispetto della norma per le gli elenchi puntati e numerati (3.1.5.4).

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 noti. 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$_G$ 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$_G$. 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$_G$ 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 usati correttamente da tutti i membri del 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$_G$ 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$_G$ dei vari prodotti parziali e finali del progetto.
Il verificatore quindi gestisce:

- La verifica e validazione$_G$ 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$_G$ 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 (*catchemallswe3@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$_G$ (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$_G$ 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 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$_G$ corrente il compito di redigere tali documenti.

4.1.5 Versionamento

GitHub$_G$ 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$_G$ che porta valore al progetto contiene il riferimento al ticket che completa (totalmente o anche solo parzialmente).

4.1.6 GitHub$_G$ Workflow

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

```
CEA-num-titolo-della-issue
```

Questo permette di identificare titolo e numero della issue$_G$ di appartenenza. Una volta fatto ciò viene creato un **primo commit$_G$**, 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$_G$ message contenente la stringa:

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

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

- Visual Studio Code;

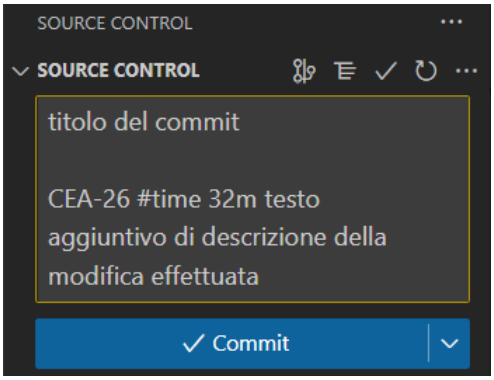

Figura 3: Immagine di come scrivere un commit su Visual Studio Code

- Git Bash.

```
git commit -m "titolo del commit\textsubscript{G}" -m "CEA-26 #time 32m ...
```

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

```
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\textsubscript{G} Workflow
```

Ciò aggiunge 1h alle ore di lavoro impiegate per la issue$_G$ con ID CEA-26, e come testo aggiuntivo per il commit$_G$ "aggiunto github Workflow", ignorato da JIRA$_G$.
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$_G$ 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$_G$ 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$_G$. Una volta revisionata la issue$_G$, se presenta qualche problema può essere spostata allo stato "in progress" dal pannello JIRA$_G$. Altrimenti attraverso una pull request nel ramo "main" e con il seguente comando posto **nel TITOLO del commit$_G$ di chiusura della pull request** la issue$_G$ verrà chiusa e considerata completata:

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

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

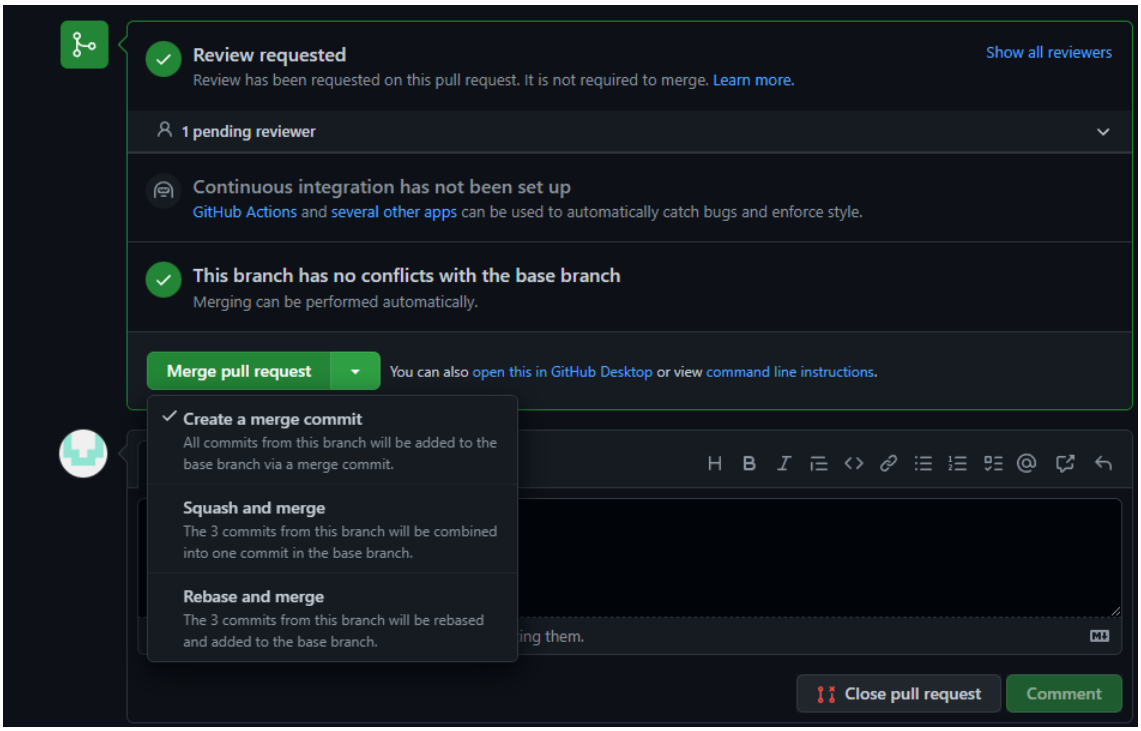

Figura 4: Merge di una pull request su Github

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

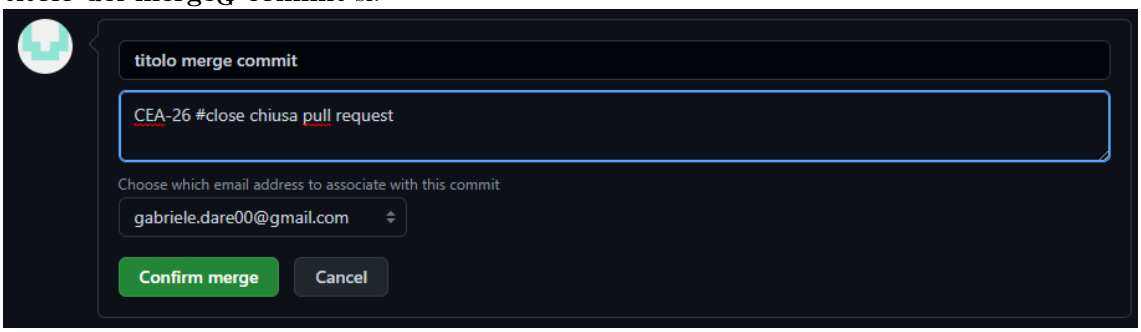

Figura 5: Messaggio esempio per effettuare il merge di una pull request

4.1.7 Issues tracking

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

- Titolo e descrizione devono, oltre ad essere sempre presenti, esplicitare 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$_G$ *Scrum$_G$* per la gestione del ciclo di sviluppo del progetto con le seguenti caratteristiche:

- Sprint$_G$ della durata di una settimana;
- Utilizzo di una board avente 4 stati. I quali sono:
  - To do;
  - In progress;
  - In review (ogni ticket deve essere validato da uno o più componenti del gruppo per essere considerato chiuso);
  - Done.

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

```
CEA–number #command <message body describing the commit\textsubscript{G}>
```

Tra i comandi troviamo:

- **Open**: permette di spostarsi da una issue$_G$ nello stadio "to do" oppure "in review" allo stadio "in progress";
- **Review**: permette lo spostamento della issue$_G$ 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".

A Standard di riferimento

A.1 Standard ISO/IEC$_{G}$ 12207

A.1.1 Scopo

La norma ha lo scopo principale di definire una struttura comune in modo che i professionisti coinvolti nello sviluppo del software (committenti, fornitori, sviluppatori, manutentori, operatori, manager e tecnici) possano utilizzare un linguaggio comune. Tale linguaggio è basato su una struttura di processi, attività, compiti e risultati prodotti. Il modello è flessibile e modulare in modo che ciascuno possa personalizzarlo a seconda delle proprie esigenze organizzative dei singoli progetti software. Lo standard stabilisce i processi presenti nel ciclo di vita del software e, per ciascuno di essi, le attività da svolgere e i risultati da produrre.

A.1.2 Tipi di Processo

I processi sono suddivisi dalla norma in tre categorie:

- **Processi primari**: I quali comprendono le attività direttamente legate allo sviluppo del software;
- **Processi di supporto**: I quali includono la gestione dei documenti e dei processi di controllo della qualità;
- **Processi organizzativi**: I quali coprono gli aspetti manageriali e di gestione delle risorse.

Per ciascun processo la norma evidenzia chiaramente:

- **Obiettivo**;
- **Responsabilità**;
- **Lista delle attività che lo compongono**;
- **Singoli compiti nei quali è suddivisa ogni attività**.

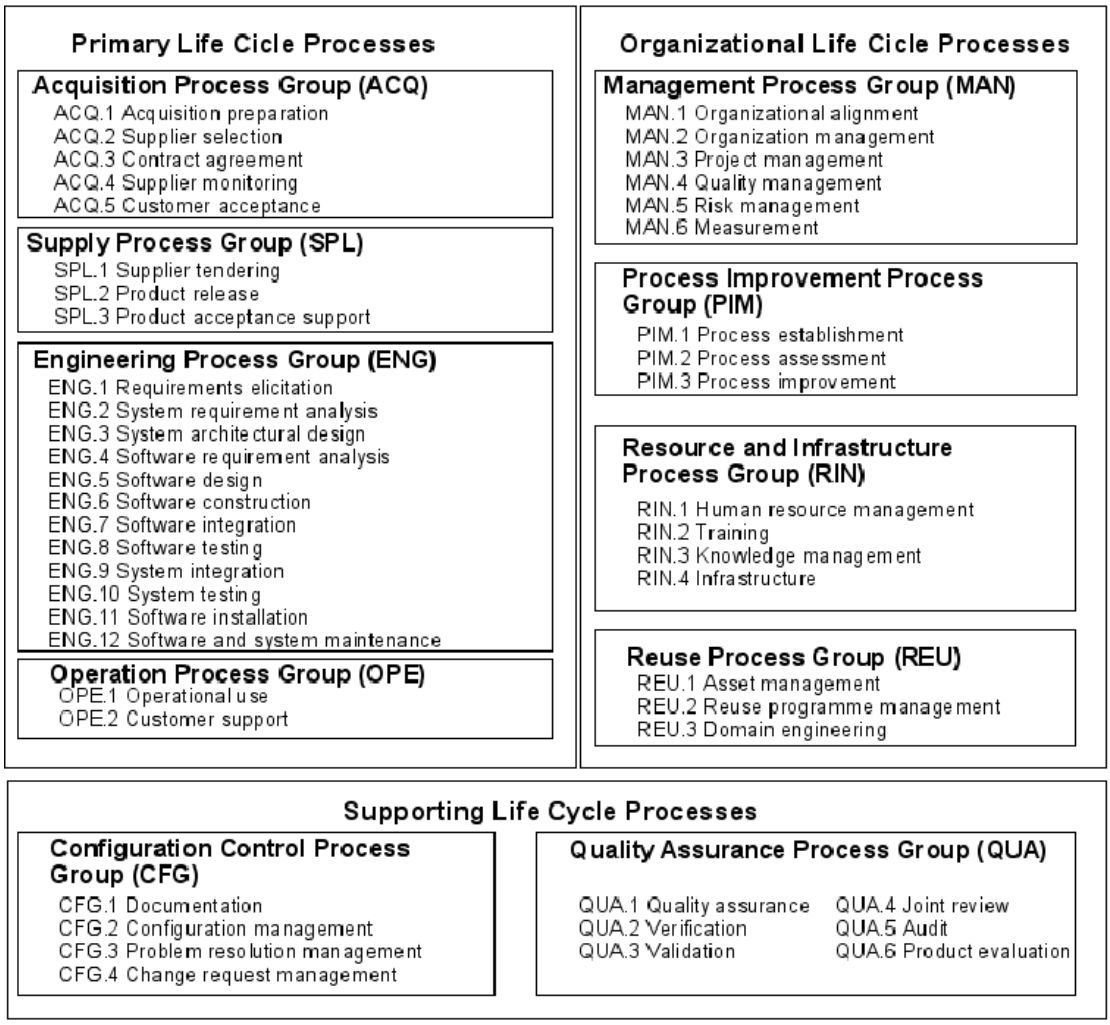

Figura 6: Processi definiti dallo standard ISO-IEC-12207

A.2 Standard ISO/IEC$_G$ 15504 SPICE

A.2.1 Scopo

Lo standard ISO/IEC$_G$ 15504, chiamato anche SPICE (Software Process Improvement and Capability Determination), fornisce un framework$_G$ per la valutazione dei processi di un'organizzazione. Questo framework$_G$ può essere utilizzato dalle organizzazioni coinvolte nella pianificazione, gestione, monitoraggio, controllo e miglioramento di acquisizione, consegna, sviluppo, implementazione, evoluzione e manutenzione di prodotti e servizi di supporto.

Dimensione del processo
La dimensione di processo del modello comprende 5 categorie di processi, rispettivamente composte da 4 a 10 processi:

- **Cliente-fornitore**: Raggruppa i processi messi in atto da un acquirente per identificare il suo bisogno, selezionare il suo fornitore e ricevere la fornitura. Dal punto di vista del fornitore, questa categoria comprende le attività necessarie per la fornitura, la messa in servizio, il funzionamento e il supporto dell'utente;
- **Engineering**: Rientrano in questa categoria le attività di sviluppo software, nell'ambito del proprio ambiente di sistema, dalla fase di definizione alla fase di manutenzione;
- **Supporto**: Raggruppa i processi che consentono l'implementazione nell'ambito di un altro processo come la documentazione o i processi di gestione della configurazione;
- **Gestione**: Questa categoria contiene i processi caratteristici delle attività di gestione, in particolare la gestione dei progetti e le attività di gestione della qualità e del rischio;
- **Organizzazione**: Questa categoria contiene i processi che riguardano l'intera organizzazione e non più il livello del singolo progetto.

Livelli di capacità e attributi di processo
Il livello di capacità della dimensione è stabilito dai seguenti 6 gradi:

- **Livello 0**: Processo incompleto o non eseguito, non raggiunge i suoi obiettivi;
- **Livello 1**: Processo svolto e implementato, gli obiettivi sono raggiunti ma non viene verificato;
- **Livello 2**: Processo gestito, la sua attuazione è pianificata, monitorata e adattata;
- **Livello 3**: Processo consolidato, si basa su pratiche documentate ed è in grado di raggiungere i propri obiettivi;
- **Livello 4**: Processo prevedibile e ripetibile, la sua attuazione è condizionata da obiettivi di performance definiti;
- **Livello 5**: Processo di ottimizzazione, per raggiungere gli obiettivi attuali e futuri, è costantemente migliorato.

La capacità di processo viene misurata utilizzando gli attributi di processo. Lo standard identifica nove attributi di processo:

- **1.1 - Prestazioni del processo**;
- **2.1 - Gestione delle prestazioni**;
- **2.2 - Gestione del prodotto di lavoro**;
- **3.1 - Definizione del processo**;
- **3.2 - Implementazione dei processi**;
- **4.1 - Misura di processo**;

- **4.2 - Controllo di processo**;
- **5.1 - Innovazione di processo**;
- **5.2 - Ottimizzazione del processo**.

In tutti i processi analizzati, vengono identificati una serie di forze e debolezze da cui si possono identificare potenziali di miglioramento. Le descrizioni del livello di maturità successivo mostrano le possibilità di miglioramento del processo. Questo standard richiede l'istituzione di una scala di valutazione. Gli attributi di ciascun processo sono valutati su una scala di valutazione suddivisa in quattro punti. I valori delle dimensioni dipendono dalla percentuale di raggiungimento degli attributi:

- **N**: Non implementato (0-15%);
- **P**: parzialmente implementato ($>$ 15-50%);
- **L**: Ampiamente implementato ($>$ 50-85%);
- **F**: completamente implementato ($>$ 85%).

A.3 Standard ISO/IEC$_{\mathbf{G}}$ 25000 SQuaRE

A.3.1 Scopo

L'ISO/IEC$_{\mathrm{G}}$ 25000 vuole dare un contributo alla sicurezza, alla funzionalità e manutenibilità del prodotto software, all'accuratezza dei dati, al raggiungimento della soddisfazione dell'utente in un'ottica preventiva e di qualità misurabile. Esso propone quindi modelli di qualità di riferimento a priori rispetto a quelli dei sistemi basati solo sulla difettosità a posteriori o monitorata durante le fasi del ciclo di vita del prodotto.

A.3.2 ISO/IEC$_{\mathbf{G}}$ 25010

Uno standard molto rilevante della serie 25000 SQuaRE e l'ISO/IEC$_{\mathrm{G}}$ 25010 il quale si occupa di definire gli obiettivi di qualità che deve avere un prodotto software. Lo standard definisce quindi i termini di:

- **Qualità interna**: La quale è riferita alle proprietà statiche e strutturali del software;
- **Qualità esterna**: La quale è riferita alle proprietà dinamiche e comportamentali del software;
- **Qualità in uso**: La quale è riferita al comportamento del software in un ambiente di utilizzo reale e alle varie interazioni con gli utenti.

Qualità interna ed esterna
I modelli di qualità interna ed esterna comprendono 8 caratteristiche a loro volta sotto-categorizzate:

- **Idoneità funzionale**: Suddivisa in:
  - **Completezza**;
  - **Adeguatezza**;

  - **Correttezza**.
- **Prestazione ed efficienza**: Suddivisa in:
  - **Tempo**;
  - **Risorse**;
  - **Capacità**.
- **Usabilità**: Suddivisa in:
  - **Riconoscibilità**;
  - **Apprendibilità**;
  - **Operabilità**;
  - **Protezione errori**;
  - **Esteticità**;
  - **Accessibilità**.
- **Affidabilità**: Suddivisa in:
  - **Maturità**;
  - **Disponibilità**;
  - **Tolleranza**;
  - **Recuperabilità**.
- **Sicurezza**: Suddivisa in:
  - **Riservatezza**;
  - **Integrità**;
  - **Non ripudio**;
  - **Autenticazione**;
  - **Autenticità**.
- **Manutenibilità**: Suddivisa in:
  - **Modularità**;
  - **Riusabilità**;
  - **Analizzabilità**;
  - **Modificabilità**;
  - **Testabilità**.
- **Compatibilità**: Suddivisa in:
  - **Coesistenza**;
  - **Interoperabilità**.
- **Portabilità**: Suddivisa in:
  - **Adattabilità**;
  - **Installabilità**;
  - **Sostituibilità**.

Qualità in uso del prodotto
I modelli di qualità in uso del prodotto comprendono 5 caratteristiche a loro volta sotto-categorizzate:

- **Efficacia**;
- **Efficienza**;
- **Soddisfazione**; Suddivisa in:
  - **Utilità**;
  - **Fiducia**;
  - **Piacere**;
  - **Comodità**.
- **Assenza e attenuazione dei rischi**: Suddivisa in:
  - **Economicità**;
  - **Salute**;
  - **Ambiente**.
- **Copertura del contesto**: Suddivisa in:
  - **Completezza**;
  - **Flessibilità**.

A.3.3 ISO/IEC$_{\mathbf{G}}$ 25023

Ad accompagnare lo standard 25010 troviamo lo standard 25023, il quale fornisce delle metriche di riferimento ai vari obiettivi di qualità di prodotto presenti nello standard a cui riferisce.