



SWEnergy

project.swenergy@gmail.com

Norme di progetto

Descrizione: Elenco delle procedure interne e delle buone pratiche di progetto adottate dal gruppo.

Stato	Non approvato
Data	21/11/2023
Verificatore	Niccolò Carlesso
Approvatore	Nessuno
Versione	1.3.0

Il responsabile: Nessuno

Registro delle modifiche

Versione	Data	Redattore	Verificatore	Approvatore	Descrizione
1.3.0	8/01/2024	Carlo Rosso	Niccolò Carlesso	Nessuno	Correzione della sotto-sezione "Aggiornamento delle "Norme di Progetto"" e aggiunte le sotto-sezioni "Revisione del codice" e "Codifica"
1.2.0	31/12/2023	Carlo Rosso	Niccolò Carlesso	Nessuno	Ristrutturazione del documento per ruolo, piuttosto che per argomento
1.1.0	30/10/2023	Carlo Rosso	Nessuno	Nessuno	Aggiornamento della sezione dedicata alla documentazione e aggiunta una sezione dedicata agli appunti
1.0.0	30/10/2023	Nessuno	Nessuno	Giacomo Gualato	Approvazione finale del documento
0.2.1	29/10/2023	Alessandro Tigani Sava	Niccolò Carlesso	Nessuno	Modifica procedure in sezione Approvazione di un documento
0.2.0	24/10/2023	Matteo Bando	Niccolò Carlesso	Nessuno	Redazione sezioni Versionamento, Verifica di un documento, Approvazione di un documento
0.1.0	23/10/2023	Alessandro Tigani Sava	Nessuno	Nessuno	Redazione sezioni Introduzione, Strumenti, Creazione e modifica di un documento, Ruoli, Registro delle modifiche

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Struttura del documento	3
2	Tutti	5
2.1	Lavoro sul progetto	5
3	Responsabile	6
3.1	Organizzare un <i>meeting</i> interno	6
3.2	Organizzare un <i>meeting</i> esterno	8
3.3	Pianificazione delle attività	9
3.4	Aggiornamento del "Piano di progetto"	11
3.5	Approvare un documento	12
4	Amministratore	13
4.1	Aggiornamento delle "Norme di progetto"	13
4.2	Aggiornamento delle "Piano di qualifica"	14
5	Verificatore	16
5.1	Verificare la correttezza di un documento	16
5.2	Revisione del codice	18
6	Analista	20
6.1	Redazione di un documento	20
6.2	Descrizione	20
7	Progettista	23
7.1	Organizzare un <i>workshop</i>	23
8	Programmatore	24
8.1	Codifica	24

1 Introduzione

1.1 Scopo del documento

Questo documento ha solo scopo di contenere le procedure e le *best practice (so far)* che SWEnergy ha deciso di adottare per sviluppare il progetto "*Easy Meal*". Dunque, serve per aiutare gli elementi di SWEnergy ad affrontare il cambio dei ruoli, in modo da rendere il passaggio più semplice e meno dispendioso in termini di tempo.

Non si considera che questo documento sia esaustivo, ma che sia sufficiente per aiutare i membri di SWEnergy a svolgere i compiti assegnati. Infatti, per agevolare lo svolgimento dei compiti assegnati, oltre alla consultazione delle "Norme di progetto", è necessario chiedere chiarimenti all'ultima persona che ha svolto il ruolo ora assegnato al lettore.

1.2 Struttura del documento

Il documento è diviso in sezioni, una per ciascun ruolo. Oggi sotto-sezione rappresenta un compito che il ruolo deve svolgere. La struttura dei compiti è la seguente:

- **Titolo;**
- **Descrizione:** riguarda la descrizione del compito, viene usata come introduzione al compito. In aggiunta, sono contenute le informazioni necessarie per lo svolgimento di qualche attività, per rendere le medesime più chiare e comprensibili;
- **Trigger:** spiega quando il compito deve essere svolto. Quindi sono elencate le condizioni che devono essere verificate per attivare il compito;
- **Scopo:** descrizione dello stato che si vuole raggiungere, in seguito al completamento del compito;
- **Svolgimento:** contiene l'elenco delle attività che il ruolo è tenuto a svolgere per completare il compito. Le attività possono essere tra loro dipendenti, oppure indipendenti. Si ritiene che il linguaggio usato sia sufficiente per rendere le attività chiare e comprensibili;

- **Attività:** per ogni attività viene fornita una breve descrizione. Se necessario, viene fornita una serie di passi da seguire per completare l'attività. I passi da seguire sono elencati in ordine e sono dipendenti tra loro.

Nel caso in cui un componente di SWEnergy abbia qualche dubbio in merito a qualche compito, è tenuto a chiedere chiarimenti.

2 Tutti

2.1 Lavoro sul progetto

2.1.1 Descrizione

Questo compito descrive i passi da seguire per svolgere qualunque altro compito assegnato dal responsabile di progetto.

2.1.2 *Trigger*

- Il responsabile di progetto assegna un compito ad un membro del gruppo.

2.1.3 Scopo

- Svolgere il compito assegnato;
- Risulta conclusa un'*issue* nella *repository* corrispondente;
- Il compito è stato verificato e convalidato da una persona diversa da chi lo ha svolto.

2.1.4 Svolgimento

Di seguito sono descritte le attività da svolgere per effettuare un compito assegnato:

- **Analisi:** si crea una *issue* nella *repository* nella quale verrà svolto il compito. La *issue* sarà assegnata a se stessi. All'interno della *issue* si descrive l'insieme degli obiettivi da raggiungere affinché l'attività sia completata. L'attività deve essere collegata al *project* corrispondente alla fase di sviluppo in cui si trova il progetto. In aggiunta, deve essere marcata con un *tag* che ne identifichi la tipologia;
- **Appunti:** si inseriscono i file degli appunti nel proprio *branch* personale all'interno della *repository* `appunti-swe`. Nella *repository*, deve essere presente un `README.md` contenente l'organizzazione della cartella per permettere agli altri membri di orientarsi;
- **Svolgimento:** si svolge l'attività assegnata, al meglio delle proprie capacità e cercando di rispettare le scadenze;

- **Verifica:** si chiede ad un membro del gruppo, tendenzialmente al verificatore, di controllare la conformità del lavoro svolto.

3 Responsabile

3.1 Organizzare un *meeting* interno

3.1.1 Descrizione

Il responsabile è tenuto ad organizzare i *meeting* interni, ovvero le *stand-up*. Le *stand-up* sono riunioni brevi, della durata di circa 30 minuti, che si svolgono su *Discord*. In esse sono trattati i seguenti argomenti:

- **Brainstorming:** i membri del gruppo riassumono brevemente il lavoro svolto nella settimana;
- **Problemi riscontrati:** i membri del gruppo espongono i problemi riscontrati durante la settimana;
- **To-do list:** sono discussi i compiti da svolgere nella settimana successiva;
- **Dubbi:** i membri del gruppo espongono i dubbi riguardo alle attività da svolgere;
- **Restrospettiva:** i membri del gruppo espongono i problemi, non inerenti alle attività, riscontrati durante la settimana e le possibili soluzioni. I problemi possono, per esempio, riguardare l'organizzazione del lavoro o la comunicazione tra i membri del gruppo o con il proponente.

3.1.2 Trigger

- Risulta necessario organizzare un *meeting* interno;

3.1.3 Scopo

- Rendere la comunicazione tra i membri del gruppo più efficace ed efficiente;
- Creare della documentazione usufruibile in caso di dubbi o problematiche;
- Formalizzare le decisioni prese durante la riunione.

3.1.4 Svolgimento

Di seguito sono riportate le attività da completare per organizzare una *stand-up*:

- **Pianificazione:** il responsabile deve decidere quando svolgere la *stand-up*. Di seguito i passi:
 1. **Anticipare la data:** nella *stand-up* precedente il responsabile si informa sulle disponibilità dei membri del gruppo per la prossima *stand-up*;
 2. **Pianificare la data:** il responsabile propone delle date e degli orari per la prossima *stand-up* e le propone sul gruppo *Telegram* del gruppo. I membri del gruppo esprimono la loro preferenza attraverso un sondaggio.
- **Ordine del giorno:** il responsabile deve stilare un ordine del giorno, ovvero una lista degli argomenti da trattare durante la riunione. Di seguito i passi:
 1. **Template:** il responsabile utilizza il template delle *stand-up* situato nella *repository* `appunti-swe`;
 2. **Brainstorming:** il responsabile si informa con i membri del gruppo attraverso *Telegram* in merito ai punti che ciascun componente di SWEnergy intende trattare durante la riunione;
 3. **To-do list:** il responsabile stila la lista delle attività da svolgere nella settimana successiva. La lista viene poi discussa e approvata durante la riunione.
- **Verbale della riunione:** il responsabile deve redigere il verbale della riunione, in cui vengono riportati gli argomenti trattati e le decisioni prese. Di seguito i passi per redigere il verbale interno:
 1. **Appunti:** l'ordine del giorno (il punto precedente) viene utilizzato come base per stilare il verbale interno;
 2. **Template:** viene copiata la cartella della riunione precedente e viene rinominata seguendo il formato: `YYYY-MM-DD_I`;
 3. **Stesura:** poichè si tratta di un documento, si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).

3.2 Organizzare un *meeting* esterno

3.2.1 Descrizione

Il responsabile è tenuto ad organizzare i *meeting* esterni, ovvero i SAL tenuti tra il proponente e SWEnergy. I SAL sono riunioni brevi, della durata di circa 30 minuti, che hanno luogo su *Teams*. In esse sono trattati i seguenti argomenti:

- **Riassunto:** il responsabile riassume le attività svolte dal gruppo durante lo sprint;
- **Problemi riscontrati:** il responsabile espone i problemi riscontrati durante lo sprint;
- **To-do list:** sono discussi i compiti da svolgere nella settimana successiva tra il gruppo e il proponente;
- **Dubbi:** il responsabile espone i dubbi riguardo alle attività da svolgere;
- **Restrospettiva:** il responsabile guida la discussione sulla qualità del prodotto e soprattutto del processo.

3.2.2 Trigger

- Risulta necessario organizzare un *meeting* interno;

3.2.3 Scopo

- Rendere la comunicazione tra i membri del gruppo più efficace ed efficiente;
- Creare della documentazione usufruibile in caso di dubbi o problematiche;
- Formalizzare le decisioni prese durante la riunione.

3.2.4 Svolgimento

Di seguito sono riportate le attività da completare per organizzare un SAL:

- **Pianificazione:** il responsabile deve decidere quando svolgere un SAL. Di seguito i passi:
 1. **Anticipare la data:** nel SAL precedente il responsabile e il proponente concordano la data del prossimo SAL;

2. **Pianificare l'ora:** il responsabile contatta su *Telegram* il proponente, gli condivide l'ordine del giorno e concorda l'ora del *SAL*. Le due attività sono svolte in concomitanza, perché si può immaginare la durata del *SAL* solo dopo aver stilato l'ordine del giorno.
- **Ordine del giorno:** il responsabile deve stilare l'ordine del giorno, ovvero una lista degli argomenti da trattare durante la riunione. Di seguito i passi:
 1. **Template:** il responsabile utilizza il template dei *SAL* precedenti, situato nella *repository* `appunti-swe`;
 2. **Brainstorming:** il responsabile si informa con i membri del gruppo attraverso le *stand-up* in merito allo *status quo* del progetto;
 3. **To-do list:** il responsabile stila la lista delle attività da svolgere nello sprint successivo. La lista viene poi discussa e approvata durante la riunione.
 - **Verbale della riunione:** il responsabile deve redigere il verbale della riunione, in cui vengono riportati gli argomenti trattati e le decisioni prese. Di seguito i passi per redigere il verbale interno:
 1. **Appunti:** l'ordine del giorno (il punto precedente) viene utilizzato come base per stilare il verbale esterno;
 2. **Template:** viene copiata la cartella della riunione precedente e viene rinominata seguendo il formato: `YYYY-MM-DD_E`;
 3. **Stesura:** poichè si tratta di un documento, si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).

3.3 Pianificazione delle attività

3.3.1 Descrizione

Il responsabile è tenuto a pianificare le attività da svolgere durante lo sprint in cui si trova il gruppo e a suddividerle tra i membri del gruppo. Inoltre deve aggiornare il programma di lavoro in base alle attività svolte e a quelle da svolgere. La pianificazione avviene tramite l'uso dei diagrammi di Gantt disponibili su *GitHub*.

3.3.2 Trigger

- Comincia una nuova iterazione, che sia uno sprint^G od un mini-sprint^G.

3.3.3 Scopo

- Aggiornare il programma di lavoro in base alle attività svolte e a quelle da svolgere.
- Pianificare le attività da svolgere durante l'iterazione corrente;
- Guidare lo svolgimento delle attività;
- Produrre la documentazione che permette di tenere traccia delle attività svolte e di quelle da svolgere.

3.3.4 Svolgimento

Di seguito sono riportate le attività da completare per pianificare le attività da svolgere:

- **Creazione delle *issue***: il responsabile deve creare delle *issue* che descrivono le attività da svolgere e guidano i lavoratori nella loro esecuzione. Di seguito sono riportati i passi per definire le *issue*:
 1. **Identificazione**: il responsabile identifica le attività da svolgere e le aggiunge su *GitHub*;
 2. **Priorità**: il responsabile assegna una priorità alle *issue* in base all'urgenza e all'importanza;
 3. **Scadenza**: il responsabile assegna una data di scadenza alle *issue* in base alla priorità e alla durata dell'attività. L'attività viene quindi inserita nel *project* di *GitHub* corrispondente alla milestone di riferimento. In questo modo viene aggiornato il diagramma di Gantt;
 4. **Perfezionamento**: il responsabile guida la discussione in merito alle *issue* durante le riunioni. In questo modo sono aggiornate la priorità, la scadenza e la descrizione;
 5. **Assegnazione**: il responsabile assegna le *issue* ai membri del gruppo in base alle loro competenze e disponibilità.

3.4 Aggiornamento del "Piano di progetto"

3.4.1 Descrizione

Il responsabile deve aggiornare il documento "Piano di progetto".

3.4.2 Trigger

- Inizio di uno sprint^G ;
- Fine di uno sprint^G ;

3.4.3 Scopo

- Formalizzare la pianificazione delle attività da svolgere durante lo sprint^G ;
- Disambiguare la pianificazione;
- Aggiornare le informazioni relative ai rischi e al modello di sviluppo;
- Aggiornare le informazioni utili alla verifica dello stato di avanzamento del progetto;

3.4.4 Svolgimento

Di seguito sono riportate le attività da completare per aggiornare il documento "Piano di progetto":

- **Rischi e modello di sviluppo:** per quanto riguarda le sezioni relative ai rischi e al modello di sviluppo, il responsabile aggiorna le informazioni in esse contenute in base all'esperienza maturata durante il periodo da responsabile;
- **Pianificazione:** il responsabile aggiorna la sezione di pianificazione rispettando la struttura già definita nel documento. Eventualmente può proporre modifiche alla struttura di pianificazione di periodo. Queste sono discusse nelle riunioni interne. Di seguito sono riportati i passi da seguire per aggiornare la sezione di pianificazione:
 1. **Creazione:** nella cartella `preventivi` viene aggiunto un nuovo file `MM_GG-P.tex` dove `MM` e `GG` indicano rispettivamente il mese e il giorno di inizio del periodo di riferimento;

2. **Stesura:** seguendo la struttura definita nei preventivi precedenti, il responsabile stila la sotto-sezione, riportando le informazioni di pianificazione relative al periodo di riferimento presenti sul progetto di *GitHub*.
- **Consuntivo:** medesimo procedimento della sezione di pianificazione. Di seguito sono riportati i passi da seguire per aggiornare la sezione di consuntivo:
 1. **Creazione** nella cartella `consuntivi` viene aggiunto un nuovo file `MM_GG-C.tex` dove MM e GG indicano rispettivamente il mese e il giorno di inizio del periodo di riferimento;
 2. **Stesura:** seguendo la struttura definita nei consuntivi precedenti, il responsabile stila la sotto-sezione, riportando le informazioni di consuntivo relative al periodo di riferimento. Nota bene: le *issue* di *GitHub* usate per tenere traccia del consuntivo sono quelle generate e chiuse dai membri del gruppo e non quelle create dal responsabile (che sono invece usate per stilare il preventivo).
 - **Modifica di un documento:** dal momento che l'aggiornamento del documento "Piano di progetto" rientra nella casistica di modifica di un documento, si rimanda alla sezione che illustra come redigere un documento (vedi 6.1).

3.5 Approvare un documento

Di seguito viene descritto il processo di approvazione di un documento.

3.5.1 *Trigger*

- Un documento viene completato.

3.5.2 **Scopo**

- Assicurarsi che il documento soddisfi i requisiti ad esso associati;
- Convalidare il contenuto ed il completamento del documento.

3.5.3 Svolgimento

Per verificare la correttezza di un documento, il responsabile deve completare le seguenti attività:

- **Approvazione:**
 1. **Seconda verifica:** il responsabile verifica il documento (vedi sottosezione 5.1);
 2. **Aggiornamento della versione:** dopo che il documento viene corretto dall'autore, il responsabile aggiorna la sua versione ed il suo stato;
 3. **Versione:** sia $X.Y.Z$ la versione del documento, dopo l'approvazione, il valore di X viene incrementato di 1, mentre Y e Z vengono azzerati.

4 Amministratore

4.1 Aggiornamento delle "Norme di progetto"

4.1.1 Descrizione

L'amministratore deve aggiornare il documento "Norme di progetto".

4.1.2 Trigger

- Si discute di qualche processo da aggiungere o modificare durante un *meeting*.

4.1.3 Scopo

- Mantenere il documento coerente rispetto al modello di sviluppo e ai processi adottati da SWEnergy;
- Formalizzare i processi adottati da SWEnergy, per chiarire eventuali dubbi e per facilitare l'individuazione di attività da svolgere;
- Mostrare l'evoluzione dell'organizzazione del lavoro di SWEnergy;
- Evidenziare i dubbi e le lacune intestini ai processi di sviluppo.

4.1.4 Svolgimento

Di seguito sono riportate le attività da completare per aggiornare il documento "Norme di progetto":

- **Identificazione delle attività:** in quale modo l'amministratore ed il gruppo possono individuare le attività da includere nel documento. Di seguito sono riportati i passi da seguire:
 1. **Nuovo compito:** durante gli incontri, SWEnergy si rende conto che alcune attività si presentano di frequente;
 2. **Ipotesi:** SWEnergy ipotizza il flusso di lavoro da svolgere per completare il compito. Sono stesi degli appunti che verranno poi inseriti nel documento "Norme di progetto";
 3. **Sperimentazione:** i componenti del gruppo che svolgono l'attività, sperimentano diverse tecniche per completare il compito, partendo dall'ipotesi iniziale;
 4. **Perfezionamento:** i componenti che hanno svolto l'attività, spiegano al gruppo il processo seguito. SWEnergy lo discute e lo valuta;
 5. **Formalizzazione:** l'amministratore inserisce il compito nel documento "Norme di progetto". Nota bene: viene modificato un documento, quindi si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).
- **Aggiornamento delle attività:** in seguito ad una discussione organica a SWEnergy, l'amministratore modifica il compito nel documento "Norme di progetto". Nota bene: viene modificato un documento, quindi si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).

4.2 Aggiornamento delle "Piano di qualifica"

4.2.1 Descrizione

L'amministratore deve aggiornare il documento "Piano di qualifica".

4.2.2 Trigger

- Termina uno sprint^G ;

4.2.3 Scopo

- Mantenere sotto controllo la qualità del prodotto;
- Mantenere sotto controllo l'andamento del progetto;
- Documentare quanto qui sopra, per poterlo mostrare al committente durante le revisioni e per evidenziarne l'evoluzione nel tempo.

4.2.4 Svolgimento

Di seguito sono riportate le attività da completare per aggiornare il documento "Piano di qualifica":

- **Identificazione di una metrica:** in quale modo l'amministratore ed il gruppo possono individuare le metriche utili a controllare e valutare la qualità del prodotto. Di seguito sono descritti i passi da seguire:
 1. **Nuova metrica:** durante gli incontri, uno dei componenti di SWEnergy propone una nuova metrica da adottare per valutare la qualità del prodotto;
 2. **Discussione:** i componenti del gruppo discutono in merito alla metrica proposta: se è utile, se è applicabile ed in quale modo verificare i risultati ottenuti e formalizzarli;
 3. **Formalizzazione:** l'amministratore inserisce la metrica di qualità nel documento "Piano di qualifica". Nota bene: viene modificato un documento, quindi si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).
- **Aggiornamento di una metrica:** In seguito ad una discussione organica a SWEnergy, l'amministratore modifica qualche caratteristica di una metrica nel documento "Piano di qualifica". Nota bene: viene modificato un documento, quindi si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).
- **Misurazione:** l'amministratore misura i risultati ottenuti applicando le metriche di qualità. Di seguito sono descritti i passi di aggiornamento del documento "Piano di qualifica". Di seguito sono riportati i passi da seguire:

1. **Nuovi risultati:** alla fine di ogni sprint, l'amministratore e il gruppo valutano i risultati di qualità ottenuti applicando le metriche concordate;
2. **Discussione:** I risultati sono discussi durante la retrospettiva e, se ritenuto opportuno, sono modificati gli obiettivi di qualità adottati da SWEnergy;
3. **Inserimento dei risultati:** l'amministratore inserisce i risultati ottenuti nel documento "Piano di qualifica". Nota bene: viene modificato un documento, quindi si rimanda alla sottosezione che illustra come redigere un documento (vedi sottosezione 6.1).

5 Verificatore

5.1 Verificare la correttezza di un documento

Il verificatore deve verificare che i documenti prodotti mentre svolge il suo ruolo siano conformi alle norme stabilite in questa sotto-sezione.

5.1.1 *Trigger*

- Viene prodotto un incremento su di un documento;
- Un componente di SWEnergy segnala la necessità di una verifica.

5.1.2 **Scopo**

- Evidenziare gli errori in un documento e segnalarli all'autore del documento;
- Assicurarsi che il documento soddisfi le norme qui sotto descritte;
- Convalidare l'incremento di un documento per garantirne l'integrità agli altri componenti di SWEnergy.

5.1.3 **Norme**

- **Correttezza grammaticale:** il testo deve essere privo di errori grammaticali;
- **Correttezza lessicale:** il testo deve essere privo di errori lessicali;

- **Correttezza ortografica:** il testo deve essere privo di errori ortografici;
- **Correttezza sintattica:** il testo deve essere sintatticamente corretto;
- **Correttezza di contenuto:** il testo deve essere privo di errori di contenuto;
- **Correttezza della struttura:** in ogni documento che contiene il registro delle modifiche, deve essere anche presente un'introduzione che spiega la struttura del documento medesimo, coerente con la struttura del documento;
- **Completezza:** il documento deve essere completo di tutte le sezioni opportune;
- **Coerenza:** il contenuto del documento deve essere coerente con il suo scopo, con le norme qui descritte e con il contenuto di eventuali documenti correlati;
- **Chiarezza espositiva:** il documento deve essere scritto in modo chiaro e comprensibile;

5.1.4 Svolgimento

Per verificare la correttezza di un documento, il verificatore deve completare le seguenti attività:

- **Correzione dei refusi:** il verificatore deve correggere i refusi presenti nel documento. Sono considerati refusi gli errori della tipologia grammaticale, lessicale, ortografica e sintattica;
- **Verifica del contenuto:** il verificatore deve verificare che il contenuto del documento sia corretto e coerente con il suo scopo. Di seguito sono riportati i passi da seguire:
 1. **Lettura del documento:** il verificatore deve leggere il documento per comprendere il contenuto del documento;
 2. **Appunti degli errori:** durante la lettura il verificatore prende nota di eventuali errori;
 3. **Ricerca delle soluzioni:** il verificatore deve trovare una soluzione per ogni errore trovato;

4. **Spiegazione degli errori:** il verificatore deve segnalare all'autore del documento gli errori trovati e le relative soluzioni;
5. **Aggiornamento della versione:** dopo che il documento viene corretto dall'autore, il verificatore deve aggiornare la versione del documento;
6. **Versione:** sia $X.Y.Z$ la versione del documento, dopo la verifica, il valore di Z viene incrementato di 1, se le modifiche apportate al documento si limitano al contenuto e non modificano la struttura del documento, ovvero l'indice non viene modificato; altrimenti il valore di Y viene incrementato di 1 e Z viene azzerato.

5.2 Revisione del codice

5.2.1 Descrizione

Il verificatore deve effettuare dei controlli di conformità sul codice prodotto. Questo controllo deve essere effettuato in modo sistematico e ripetitivo.

5.2.2 Trigger

- Viene prodotto un incremento sulla *code base*;
- Un componente di SWEnergy segnala la necessità di una verifica.

5.2.3 Scopo

- Evidenziare gli errori nel codice e segnalarli al programmatore;
- Assicurarsi che il codice soddisfi le norme qui sotto descritte;
- Convalidare l'incremento di codice per garantirne l'integrità agli altri componenti di SWEnergy

5.2.4 Norme

- **Commenti:** per ciascuna funzione o metodo, deve essere spiegato lo scopo. In particolare, deve essere sempre presente la spiegazione dei parametri in ingresso e del valore di ritorno;

- **Test:** per ciascuna funzione o metodo, deve essere presente almeno un test che ne verifica il corretto funzionamento e fornisce un esempio di utilizzo;
- **Nomi:** i nomi delle variabili devono essere significativi e devono essere scritti in lingua italiana. Di seguito sono riportate le regole di forma per ciascun tipo di variabile:
 - **Variabili:** devono essere scritte in minuscolo e devono essere separate da underscore (es. nome_variabile);
 - **Costanti:** devono essere scritte in maiuscolo e devono essere separate da underscore (es. NOME_COSTANTE);
 - **Interfacce:** la prima lettera di ogni parola è maiuscola e le parole sono unite senza spazi (es. NomeInterfaccia);
 - **Classi:** la prima lettera di ogni parola è maiuscola e le parole sono unite senza spazi (es. NomeClasse);
 - **Metodi:** devono essere scritte in minuscolo e devono essere separate da underscore (es. nome_metodo);
 - **Funzioni:** devono essere scritte in minuscolo e devono essere separate da underscore (es. nome_funzione);
 - **File:** devono essere scritte in minuscolo e devono essere separate da underscore (es. nome_file). In ogni *file* ci può essere al più una classe o un'interfaccia che ha lo stesso nome del *file*.

5.2.5 Svolgimento

Di seguito sono riportate le attività da completare per effettuare i controlli di conformità sul codice prodotto:

- **Correzione del codice:** il verificatore deve controllare che per ciascuna funzione o metodo sia presente una descrizione dello scopo, dei parametri in ingresso e del valore di ritorno. Di seguito sono riportati i passi da seguire:
 1. **Commenti:** il verificatore legge i commenti della funzione e ne intuisce lo scopo;
 2. **Funzionamento:** il verificatore legge il corpo della funzione o del metodo e ne verifica il funzionamento staticamente;

3. **Test:** il verificatore verifica che sia presente almeno un test per la funzione o il metodo;
4. **Edge cases:** il verificatore verifica che i test siano completi e che coprano tutti i casi particolari;
5. **Nomi:** il verificatore controlla che i nomi definiti dal programmatore rispettino le regole di forma definite precedentemente;
6. **Correzioni:** il verificatore riporta gli errori riscontrati al programmatore;
7. **Aggiornamento della versione:** dopo che il codice viene corretto dal programmatore, il verificatore deve aggiornare la versione del codice;
8. **Versione:** sia $X.Y.Z$ la versione del codice, dopo la verifica, il valore di Z viene incrementato di 1, se le modifiche apportate al codice non aggiungono nuove funzionalità. Se invece le modifiche apportate al codice aggiungono nuove funzionalità, il valore di Y viene incrementato di 1 e il valore di Z viene reimpostato a 0. Una funzionalità coincide con un requisito.

6 Analista

6.1 Redazione di un documento

6.2 Descrizione

L'analista redige i documenti, in particolare l'analista redige l'"Analisi dei requisiti".

6.2.1 *Trigger*

- Sono presenti dei dubbi o delle lacune in merito a qualcosa;
- Risulta necessario formalizzare qualche concetto o qualche argomento.

6.2.2 **Scopo**

- Risolvere i dubbi e le lacune riguardo a un argomento, o almeno formalizzare i dubbi e le lacune;

- Formalizzare la definizione di un concetto o di un argomento, per renderlo chiaro ed inequivoco.

6.2.3 Strumenti

Gli strumenti utilizzati per la creazione dei documenti sono:

- **LaTeX**: linguaggio di *markup* per la creazione di documenti (www.latex-project.org);
- **VisualStudio Code**: GUI con integrazioni per la creazione di documenti scritti in LaTeX e per la gestione delle repository git (code.visualstudio.com)
 - **LaTeX Workshop**: estensione utilizzata in VisualStudio Code per la compilazione e la scrittura dei documenti.

6.2.4 Struttura del documento

A ciascun documento corrisponde un'omonica cartella che viene creata all'interno della cartella che rappresenta la fase in cui si trova il progetto quando viene prodotto il documento. La cartella della fase si trova nella *repository* `doc-latex` dell'organizzazione GitHub del gruppo. Il nome della cartella è il nome del documento in deve avere la prima lettera maiuscola; sono previsti gli spazi tra le parole e le parole successive alla prima sono in minuscolo. Di seguito la struttura della cartella:

```
/ (Nome del documento)
├── main.tex
├── sec
│   ├── registro_modifiche.tex
│   ├── introduzione.tex
│   └── le_altre_sezioni.tex
```

6.2.5 main.tex

Di seguito la struttura del file `main.tex`:

- **Import dei template**: sono importati i template per la creazione del documento. I template sono: `copertina.tex`, `header_footer.tex` e `variable.tex`. In aggiunta, sono importati i modelli specifici per il documento che si sta redigendo;

- **Inizializzazione delle variabili:** sono inizializzate le variabili che verranno utilizzate nel documento;
- **Struttura del documento:** attraverso l'uso degli `input` viene gestita la struttura del documento.

6.2.6 Svolgimento

Di seguito sono elencate le attività che l'analista deve svolgere per la redazione di un documento:

- **Modifica di un documento:** l'analista aggiorna il documento in base alle modifiche richieste dal verificatore e in base alle informazioni necessarie per la redazione del documento. Di seguito sono elencati i passi per completare l'attività:
 1. **Pull:** l'analista effettua il *pull* della *repository* `doc-latex` per avere l'ultima versione della *repository*;
 2. **Checkout:** l'analista effettua il *checkout* del *branch* verso il *branch* chiamato come il documento che si sta redigendo;
 3. **Struttura:** l'analista modifica il `main.tex` in base alle modifiche necessarie;
 4. **Gestione dei file:** l'analista crea, elimina o rinomina i *file* nella cartella `sec` in modo tale che siano rispecchiate le modifiche apportate al `main.tex`;
 5. **Contenuto:** l'analista modifica i *file* nella cartella `sec` in base alle modifiche necessarie;
 6. **Push:** l'analista effettua un *commit* e il *push*;
 7. **Pull request:** l'analista può creare una *pull request* verso il `main`, per chiedere al verificatore, la verifica del documento;
 8. **Verifica:** l'analista informa il verificatore che il documento è pronto per la verifica;
 9. **Correzione:** l'analista corregge il documento in base alle segnalazioni del verificatore;
 10. **Chiusura:** l'analista effettua un *push* del *branch* inserendo nel messaggio di *commit* la parola `close` seguita dal numero della issue che si sta risolvendo;

11. **Secondo merge:** l'analista può concludere la *pull request* con il *main*.

7 Progettista

7.1 Organizzare un *workshop*

7.1.1 Descrizione

I progettisti sono tenuti a sperimentare con nuove tecnologie, per produrre le PoC. SWEnergy non norma il processo di sperimentazione e di produzione delle PoC, d'altra parte, ritiene che sia importante spiegare i risultati ottenuti dalle PoC al resto del *team*. I *workshop* sono un'insieme di appunti, di presentazioni e di codice per illustrare i risultati ottenuti dalle PoC.

7.1.2 Trigger

- Qualche membro del gruppo non conosce qualche tecnologia da implementare.

7.1.3 Scopo

- Condividere le conoscenze tecniche tra i membri del gruppo;
- Documentare le conoscenze tecniche acquisite;
- Imparare ad usare la nuova tecnologia;
- Provvedere affinché tutti i membri di SWEnergy abbiano una conoscenza di base e sufficienti per adottare la tecnologia all'interno del progetto.

7.1.4 Svolgimento

Di seguito sono elencate le attività che i progettisti devono svolgere per organizzare un *workshop*:

- **Bozza di appunti:** i progettisti sono tenuti a produrre dei *markdown* per spiegare e riassumere i contenuti delle PoC. I *file* così prodotti sono organizzati come il progettista meglio crede, all'interno del *repository* `appunti-swe`.

- **Appunti web:** a partire dagli appunti sopra prodotti, sono organizzati i *workshop*. Di seguito sono elencati i passi da seguire per pubblicare gli appunti di un *workshop*:
 1. Creare una cartella all'interno del *repository* `Project-SWEnergy.github.io` con il nome del *workshop* da organizzare;
 2. Creare un `readme.md` all'interno della cartella appena creata, che collega gli appunti all'interno della cartella tra loro;
 3. Effettuare il *push* delle modifiche sul *repository* remoto.
- **Presentazione:** i progettisti sono tenuti a produrre una presentazione per presentare gli appunti prodotti e le PoC realizzate. Si noti che la presentazione non ha una descrizione prescrittiva perché, a seconda del contenuto e delle conoscenze tecnologiche del progettista, può essere realizzata con diversi strumenti. Viene consigliato l'uso di *Obsidian* e del *plugin Advanced Slides*.

8 Programmatore

8.1 Codifica

8.1.1 Descrizione

Il programmatore scrive il codice sorgente che compone l'applicativo. Il codice sorgente è scritto in linguaggio TypeScript.

8.1.2 Trigger

- Viene completata la progettazione di una *feature*;

8.1.3 Scopo

- Implementare le funzionalità richieste dal proponente;
- Soddisfare qualche requisito;

8.1.4 Svolgimento

Di seguito sono elencate le attività che il programmatore deve svolgere per la codifica del prodotto software:

- **Progettazione:** il programmatore deve produrre dei commenti o degli appunti che descrivano la struttura del codice che andrà a scrivere nella prossima attività. Questi commenti devono poi essere riorganizzati e riportati nella *issue* corrispondente;
- **Test:** il programmatore deve scrivere dei test per verificare il corretto funzionamento del codice che andrà a scrivere;
- **Codifica di una funzione o metodo:** di seguito sono elencati i passi che il programmatore deve seguire per la codifica del prodotto software:
 1. **Pull:** il programmatore esegue un *pull* del codice sorgente dal *repository* remoto;
 2. **Branch:** il programmatore crea un nuovo branch di lavoro a partire dal branch *dev*;
 3. **Commenti:** il programmatore scrive lo scopo della funzione o del metodo che andrà a codificare e ne descrive la firma;
 4. **Codifica:** il programmatore scrive il codice che compone il corpo della funzione o del metodo;
 5. **Test:** il programmatore esegue i test di verifica. In caso di fallimento, il programmatore deve correggere il codice e ripetere la verifica;
 6. **Iterazione:** se il programmatore vuole scrivere altre funzioni torna al punto 3, altrimenti prosegue con il punto successivo;
 7. **Push:** il programmatore esegue un *push* del codice sorgente sul *repository* remoto.
 8. **Verifica:** il programmatore segnala al verificatore che il codice è pronto per essere verificato.
 9. **Correzione:** se il verificatore segnala degli errori, il programmatore deve correggere il codice e torna al passo precedente. Altrimenti, il programmatore può procedere al passo successivo.

10. **Chiusura:** il programmatore effettua il merge del branch di lavoro con il branch `dev` e chiude il ticket di *GitHub* corrispondente.