



Norme Di Progetto

2024-11-21

V0.0.3

sweetenteam@gmail.com

<https://sweetenteam.github.io>



Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin
Redattori	Matteo Campagnaro Orlando Ferazzani
Verificatori	Matteo Campagnaro Orlando Ferazzani

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
0.0.3	2024-11-21	Orlando Ferazzani	Matteo Campagnaro	Stesura del capitolo: Processi Organizzativi - Gestione dei Processi
0.0.2	2024-11-21	Matteo Campagnaro	Orlando Ferazzani	Stesura del capitolo: Introduzione
0.0.1	2024-11-20	Orlando Ferazzani	Matteo Campagnaro	Stesura del capitolo: Processi di Supporto - Gestione della qualità

Indice

1) Introduzione	5
1.1) Scopo del documento	5
1.2) Scopo del progetto	5
1.3) Glossario	5
1.4) Riferimenti	5
1.4.1) Riferimenti normativi	5
1.4.2) Riferimenti informativi	5
2) Processi di Supporto	6
2.1) Gestione della qualità	6
2.1.1) Scopo	6
2.1.2) Descrizione	6
2.1.3) Piano Di Qualifica	6
2.1.4) Testing	6
2.1.5) Metriche	7
2.1.6) Aspettative	7
3) Processi Organizzativi	7
3.1) Gestione dei Processi	7
3.1.1) Scopo	7
3.1.2) Descrizione	7
3.1.3) Pianificazione	8
3.1.3.1) Scopo	8
3.1.3.2) Descrizione	8
3.1.3.3) Aspettative	8
3.1.3.4) Ruoli	8
3.1.3.5) Ticketing	9
3.1.4) Coordinamento	10
3.1.4.1) Comunicazioni	10
3.1.4.1.1) Interne	10
3.1.4.1.2) Esterne	11
3.1.4.2) Riunioni	11
3.1.4.2.1) Interne	11
3.1.4.2.2) Esterne	11
3.1.4.3) Verbali	11
3.1.4.3.1) Interni	11
3.1.4.3.2) Esterni	11

Lista della immagini

Lista delle tabelle

1) Introduzione

1.1) Scopo del documento

Il presente documento ha lo scopo di definire le linee guida operative che il gruppo adotterà per la realizzazione del progetto didattico. In questo documento sono raccolte le procedure da seguire per ogni *processo* e attività. Per la realizzazione di questo documento è prevista la modifica e l'aggiunta di nuove indicazioni in base alle decisioni che verranno prese dal gruppo durante lo svolgimento del progetto. Tutti i membri del gruppo si impegnano a consultare regolarmente questo documento e ad attenersi scrupolosamente alle procedure in esso descritte, al fine di garantire un approccio professionale, coerente e uniforme nello svolgimento delle attività.

1.2) Scopo del progetto

Il team AzzurroDigitale utilizza quotidianamente diverse piattaforme per redigere documentazione e consultare informazioni essenziali per i progetti, questo può spesso comportare inefficienze. L'obiettivo del progetto Buddybot è lo sviluppo di una piccola piattaforma web con un'interfaccia *chat* per interagire con l'*IA* che funga da assistente virtuale. Questo deve essere in grado di ottenere in modo facile e veloce informazioni dalle fonti specificate e di fornirle in base alle domande poste tramite *chat* in linguaggio naturale. Tali informazioni devono essere aggregate e centralizzate da diverse fonti tra cui *GitHub*, *Confluence* e *Jira* permettendo un accesso facile e immediato con il fine di migliorare la produttività e dare supporto all'*OnBoarding*.

1.3) Glossario

Per garantire chiarezza e coerenza nella terminologia utilizzata nei documenti, è stato deciso di creare un glossario contenente le definizioni dei termini. Questo strumento raccoglierà tutti i termini specifici del dominio d'uso, accompagnati dai rispettivi significati. L'inclusione di un termine nel glossario sarà segnalata attraverso *questo stile* dedicato.

1.4) Riferimenti

1.4.1) Riferimenti normativi

- Norme di Progetto v1.0.0
- Documentazione e presentazione del capitolato d'appalto C9: BuddyBot
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C9.pdf> (Ultimo accesso: 2024-11-21)
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C9p.pdf> (Ultimo accesso: 2024-11-21)
- Regolamento del progetto didattico:
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf> (Ultimo accesso: 2024-11-21)

1.4.2) Riferimenti informativi

- I processi di ciclo di vita del software
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T02.pdf> (Ultimo accesso: 2024-11-21)
- Glossario v1.0.0:
<https://sweetenteam.github.io/docs/RTB/Glossario>
- Documentazione *git*:
<https://git-scm.com/docs> (Ultimo accesso: 2024-11-21)
- Documentazione *GitHub*:
<https://docs.github.com/en> (Ultimo accesso: 2024-11-21)

- Documentazione *Typst_G*
<https://typst.app/docs/> (Ultimo accesso: 2024-11-21)

2) Processi di Supporto

2.1) Gestione della qualità

2.1.1) Scopo

Il processo di Gestione Della Qualità ha come obiettivo principale quello di garantire che i processi, i prodotti e gli artefatti tutti nel ciclo di vita del progetto, siano conformi ai piani specificati, allineati con i requisiti stabiliti e che, in generale, rispettino i canoni di qualità in precedenza decisi.

2.1.2) Descrizione

Per garantire e mantenere un certo standard di qualità, il team si impegna ad utilizzare altri processi di supporto, come ad esempio la *Verifica_G*, *Validazione_G* e revisioni aggiuntive con il *Proponente_G*. Una volta stabiliti e definiti gli standard da rispettare nel *Piano Di Qualifica_G*, rimane solamente da assicurarsi che questi vengano applicati e rispettati in ogni fase del progetto. Sono state quindi definite le best practices di modo che ogni componente del gruppo lavori seguendo le stesse direttive e linee guida. In questo modo, applicare e rispettare questi standard diventa più semplice e veloce. Per controllare che questi standard siano applicati durante tutta la durata del progetto, ogni prodotto o artefatto è controllato da un *Verificatore_G*.

2.1.3) Piano Di Qualifica

Il documento **Piano di Qualifica** è fondamentale per garantire il rispetto gli standard di qualità richiesti, per il completamento degli obiettivi di questo *processo_G* e per soddisfare le richieste e le aspettative degli *Stakeholders_G*.

Il documento si estende, e non si limita, a:

- Definire gli obiettivi di qualità del progetto;
- Definire le metriche di visione quantitativa;
- Definire test di qualità e di accettazione;
- Fornire una retrospettiva sul progetto;

2.1.4) Testing

Il suddetto documento **Piano di Qualifica** definisce obiettivi di qualità sia dei processi che dei prodotti. Le metriche relative forniscono una verifica sugli aspetti di accessibilità, mentre i test garantiscono la qualità generale del software e dei processi. Le categorie di test sono:

1. *Test di unità_G*: verifica il corretto sviluppo e funzionamento di un'unica unità del sistema. Un'unità è definita come un elemento indivisibile e unico del sistema stesso.
2. *Test di integrazione_G*: verifica il corretto funzionamento di più unità del sistema, integrate tra loro, ma che cooperano per svolgere un'unica funzione all'interno del sistema.
3. *Test di sistema_G*: verifica il funzionamento del sistema nel suo intero. I requisiti funzionali, di vincolo, di qualità e di prestazione precedentemente concordati con il Proponente sono testati e verificati in questa fase.
4. *Test di accettazione_G*: verifica il soddisfacimento del Proponente rispetto ai requisiti concordati. Questi test sono eseguiti in presenza del Proponente stesso. Il superamento di questi test garantisce il rilascio del prodotto.

2.1.5) Metriche

Le metriche di qualità sono fondamentali per garantire il rispetto degli standard di qualità richiesti. Il documento **Piano di Qualifica** fornisce varie metriche utilizzate per misurare e rispettare gli standard di qualità. Le metriche sono divise in due categorie:

- Processo: PRC
- Prodotto: PRD

Inoltre, le suddette metriche sono identificate come segue:

M-[Categoria-Metrica]-[Sigla-Identificativa-Metrica]

Ogni metrica ha:

- un valore minimo (o accettabile): valore sotto il quale il processo o il prodotto misurato diventa inaccettabile
- un valore ammissibile (o ottimale): valore ideale che dovrebbe essere raggiunto dalla metrica
- descrizione: descrizione breve della metrica

2.1.6) Aspettative

Il team si aspetta di rispettare in ogni momento gli standard di qualità definiti. Inoltre, il team si aspetta:

- Ottima qualità del prodotto
- Ottima qualità dei processi
- Miglioramento costante
- Test frequenti e predicibili
- Soddisfazione delle richieste e aspettative del Proponente

3) Processi Organizzativi

I processi organizzativi sono fondamentali per garantire il corretto svolgimento del progetto in conformità con gli standard di qualità prefissati.

3.1) Gestione dei Processi

3.1.1) Scopo

Come stabilito dallo Standard ISO/IEC 12207:1997, il processo organizzativo di gestione dei processi ha come scopo quello di identificare le attività generali che ogni membro del team è chiamato a svolgere. E' un'attività chiave per garantire il completamento del progetto in modo efficiente e in linea con gli standard di qualità già definiti e con le aspettative del Proponente.

3.1.2) Descrizione

Il processo è diviso nelle seguenti attività:

- Inizio e definizione dello Scopo;
- Pianificazione;
- Esecuzione e Controllo;
- Revisione e Valutazione;
- Chiusura.

3.1.3) Pianificazione

3.1.3.1) Scopo

Come da Standard ISO/IEC 12207:1997, il **Responsabile_G** ha il compito di predisporre i piani per l'esecuzione di tutte le attività di pianificazione. I piani dovranno essere dettagliati, contenere la descrizione delle attività e dei compiti (e ruoli) assegnati. Il Responsabile dovrà redigere questa pianificazione all'interno del documento **Piano di Progetto_G**. Questo documento riporta tutte le descrizioni e pianificazioni delle attività da svolgere in un certo periodo (uno o più sprint)

3.1.3.2) Descrizione

L'attività di pianificazione viene articolata nelle due seguenti sezioni:

1. Ruoli
2. Ticketing

3.1.3.3) Aspettative

L'attività di pianificazione aiuta il team a:

- Organizzare e pianificare le attività da svolgere;
- Assegnare i compiti ai membri del team;
- Rispettare le scadenze prefissate;
- Monitorare e controllare l'avanzamento del progetto.

Inoltre serve al Team per rispettare le regole organizzative stabilite per lo svolgimento del progetto.

3.1.3.4) Ruoli

I ruoli sono decisi dal Responsabile di Progetto. Questa figura è coordinatrice di tutti i membri, e delle attività da essi svolte. Al termine del progetto, ogni membro dovrà aver ricoperto ogni ruolo. **SweeTenTeam** ha deciso di ruotare i ruoli ogni sprint (ogni due settimane).

I ruoli in depth:

Responsabile di progetto

Ha il compito fondamentale di rappresentare il gruppo e di esporlo al Proponente e ai committenti. I compiti di questo ruolo sono:

- Approvare la documentazione
- Gestire la pianificazione del progetto
- Coordinare i membri del gruppo assegnando tasks e ruoli
- Studiare e gestire l'analisi dei rischi
- Coordinare e curare i rapporti esterni ed interni.

Amministratore

Definisce, controlla e gestisce l'ambiente e gli strumenti di lavoro del progetto. Ha piena responsabilità sull'efficacia ed efficienza del **Way of Working_G**

I principali compiti sono:

- Redigere e mantenere aggiornata la documentazione: gestisce il versionamento
- Gestire la configurazione del progetto e del prodotto: controllo sul prodotto software
- Gestire i processi: problem solving rispetto ai processi
- Migliorare l'ambiente di lavoro: ricerca gli strumenti e le tecnologie necessarie per migliorare e automatizzare il lavoro

Analista

L' **Analista_G** approfondisce le richieste, tecniche e non, del **Capitolato_G**. E' principalmente fondamentale, e presente, nella prima fase del progetto, quando viene redatta l'analisi dei requisiti. Supervisiona la scrittura della stessa, in quanto è di vitale importanza che non vi siano errori. Avere errori nell'Analisi dei Requisiti comprometterebbe l'intero progetto. Ha il compito di:

- Studiare il progetto e il suo contesto applicativo
- Raccogliere, analizzare e studiare i bisogni del cliente
- Scrivere l'Analisi dei Requisiti
- Definire la complessità dei vari requisiti

Progettista

Determina le scelte tecniche del progetto, trasformando i requisiti trovati dagli Analisti in un'architettura che strutturi e rappresenti il problema. Il progettista inoltre seguirà lo sviluppo ma non la manutenzione.

I compiti principali sono:

- Sviluppare un prodotto economico e facilmente manutenibile partendo dalle indicazioni degli Analisti
- Favorire efficienza ed efficacia con le sue scelte

Verificatore

Controlla il lavoro svolto dagli altri membri del Team e si assicura che sia conforme agli standard di qualità definiti.

Il Verificatore ha il compito di:

- Verificare che le attività svolte siano corrette e conformi agli standard attraverso le tecniche e gli strumenti definiti nelle Norme di Progetto

Programmatore

Svolge l'attività di codifica del progetto e delle sue componenti di supporto basandosi sull'architettura indicata dal Progettista.

I compiti principali sono:

- Implementare le funzionalità richieste dal Progettista
- Scrivere codice pulito, documentato e manutenibile che rispetti le Norme di Progetto
- Creare Test per verifica e validazione del codice
- Scrivere il **manuale utente_G**

3.1.3.5) Ticketing

Il Team utilizza il sistema di **Issue Tracking Sistem_G** (ITS) di GitHub. Questo permette di assegnare compiti e ruoli ai membri del Team, di monitorare l'avanzamento del progetto e di tenere traccia delle attività svolte, oltre ad essere facilmente collegabile a **Pull Request_G** e **Commit_G**.

Le issue principali sono create dal responsabile del gruppo che assegna i compiti ai vari membri del team. In seguito, ogni problema è ulteriormente suddiviso in sotto-problemi e di conseguenza, «sotto-issue». Queste ultime sono create non dal Responsabile, ma dall'assegnatario della issue principale. Questo permette di avere una visione chiara e dettagliata delle attività da svolgere e di monitorare l'avanzamento del progetto. Ad esempio:

- Responsabile crea la Issue Norme di Progetto e la assegna al componente X
- X studia il documento Norme di progetto e individua sotto-issue per rendere il problemapiù avvicinabile.
- X crea le sotto-issue individuate, le auto-assegna e sceglie il verificatore.

Ogni Issue è composta da:

- Titolo: titolo della issue
- Descrizione: descrizione dell'attività da completare
- Assegnatario (o Assegnatari): chi deve svolgere questa issue
- Verificatore: chi deve verificare il lavoro svolto
- **Milestone_G**: traguardo da raggiungere
- **Label_G**: etichetta per categorizzare le issue
- **Project_G**: progetto a cui appartiene la issue
- stato: aperta, chiusa, in corso, ecc.
- data di inizio: data di inizio dell'attività
- data di fine: data di fine dell'attività

La creazione e la gestione della issue segue il seguente flusso:

1. Creazione della Issue
2. Assegnazione della Issue
3. Creazione delle sotto-issue
4. Assegnazione delle sotto-issue
5. Inizio dell'attività
6. Spostamento della Issue da To Do a In Progress
7. Svolgimento attività
8. Fine attività
9. Creazione Pull Request
10. Verifica del lavoro svolto
 1. se corretto, si passa al punto 11
 2. se non corretto, il verificatore lascia commenti sulla PR riguardanti le modifiche necessarie, l'assegnatario della issue apporta le modifiche e si ripete il punto 9
11. Chiusura della Issue
12. Spostamento della Issue da In Progress a Done
13. Chiusura PR e merge su branch Develop
14. Eventuale eliminazione del branch (solo se non deve essere riutilizzato)

Milestones

Il gruppo, dopo una pianificazione accurata, ha deciso di impostare le milestones ad «artefatti» e non a «sprint». Il gruppo ritiene che le Milestones indichino qualcosa di completato, quindi è stato deciso di ritenere una milestone completata solamente quando un artefatto è stato completato e verificato. Al contrario, è stata impostata una data di inizio e una data di fine alla singola issue, di modo che sia comunque monitorabile nel tempo l'avanzamento del progetto.

3.1.4) Coordinamento

E' l'attività responsabile della gestione delle comunicazioni e incontri, sia interni che esterni. Questo compito assume un ruolo fondamentale nella corretta riuscita del progetto, garantendo l'efficienza e l'efficacia del Team stesso.

Le attività di coordinamento comprendono le comunicazioni interne ed esterne, l'organizzazioni delle riunioni e la gestione delle scadenze.

3.1.4.1) Comunicazioni

3.1.4.1.1) Interne

Le comunicazioni interne avvengono con due modalità: le comunicazioni scritte, formali e non, avvengono attraverso Telegram. Le comunicazioni verbali, quindi le riunioni, la condivisione di file, di codice, e di qualsiasi altra risorsa avvengono attraverso Discord.

Su Telegram sono affrontate solamente questioni di rapida risoluzione, e comunicazioni off-topic. Le decisioni più strutturate, come elencato sopra, verranno prese su discord in presenza di tutti i membri del Team durante la riunione settimanale.

3.1.4.1.2) Esterne

Tutte le comunicazioni esterne sono gestite dal Responsabile del Progetto, previa consultazione con il Team e visione della comunicazione stessa. Vengono effettuate esclusivamente tramite email attraverso questo indirizzo: sweetenteam@gmail.com.

3.1.4.2) Riunioni

Il responsabile attuale ha il compito di introdurre i punti che saranno trattati nella riunione e di redigere il verbale interno, oltre che a decidere chi lo verificherà. La riunione avviene su Discord, in un canale dedicato.

3.1.4.2.1) Interne

Le riunioni interne avvengono settimanalmente, il giorno e l'ora sono decisi dal Responsabile e comunicati al Team in base alle esigenze di ogni membro. Queste riunioni servono per discutere lo stato del progetto, le attività svolte e da svolgere, e per prendere decisioni importanti.

3.1.4.2.2) Esterne

Le riunioni esterne avvengono con il Proponente e con i Committenti ad orari prestabiliti: i martedì dalle 17 alle 18. Le riunioni esterne sono effettuate su Google Meet. Queste riunioni sono gestite dal Responsabile del Progetto, che si occupa di organizzare l'incontro, di redigere l'ordine del giorno e di redigere il verbale esterno. Quest'ultimo viene condiviso con il Team e con il Proponente.

3.1.4.3) Verbali

Dopo ogni incontro, sia esso interno od esterno, viene redatto un verbale dal Responsabile e verificato dal Verificatore stabilito.

3.1.4.3.1) Interni

In ogni incontro interno vengono affrontati diversi temi, come si vede al punto 3.1.4.2.1. Al termine di una riunione, viene creata dal responsabile una issue, che è auto-assegnata. Dopo la redazione di tale verbale, viene verificato, creata la PR e chiusa la issue.

3.1.4.3.2) Esterni

Le modalità sono le stesse del punto precedente. Dopo che la issue viene chiusa, il verbale viene mandato al Proponente e ai Committenti per essere firmato e approvato. Dopo la firma, il verbale viene caricato su GitHub.