



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/26



Gruppo 17

Nome: BitByBit

Email: swe.bitbybit@gmail.com

Norme di progetto

Registro delle modifiche

Versione	Data	Autore	Descrizione	Verificatore
0.1.1	2025-12-02	Riccardo Manisi	Sistematì errori lessicali	Marco Sanguin
0.1.0	2025-12-01	Riccardo Manisi	Creazione del documento. Aggiunto processo di documentazione e bozza per le sezioni successive.	Marco Sanguin

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	5
1.4.1	Riferimenti normativi	5
1.4.2	Riferimenti informativi	5
2	Processi primari	5
3	Processi di supporto	5
3.1	Documentazione	5
3.1.1	Obbiettivi	6
3.1.2	Strumenti a supporto	6
3.1.3	Tipologie di documenti	7
3.1.4	Implementazione di processo	7
3.1.4.1	Creazione delle issue GitHub	7
3.1.4.2	Assegnazione della issue	7
3.1.4.3	Impostazione del documento	7
3.1.4.4	Date	10
3.1.4.5	Modifiche a documenti	11
3.1.4.6	Conclusione modifiche	12
3.1.4.7	Verifica	12
3.1.4.8	Approvazione e pull request	12
3.1.4.9	Pubblicazione	12
3.2	Processo di gestione della configurazione	12
4	Processi organizzativi del ciclo di vita	12
4.1	Gestione dei processi	12

1. Introduzione

1.1. Scopo del documento

Il presente documento è volto a definire i processi del way of working per il gruppo BitByBit che si impegna a mantenerlo per tutta la durata del progetto. Per la sua redazione è stato preso come riferimento lo Standard ISO/IEC 12007:1995 che identifica 3 tipologie di processo:

1. **processi primari:** atti alla produzione di un prodotto software;
2. **processi di supporto:** supportano altri processi come parte integrante e contribuiscono al successo e alla qualità del prodotto software;
3. **processi organizzativi:** definiscono i metodi organizzativi del gruppo, per stabilire e implementare la struttura costituita dai processi di ciclo di vita.

Ogni membro del gruppo si impegna a visionare costantemente il presente documento e a rispettare rigorosamente i processi che vengono esposti di seguito, per ottenere un prodotto che sia professionale, coerente e uniforme.

1.2. Scopo del prodotto

L'azienda **Miriade srl** ha proposto lo sviluppo di un'applicazione mobile, denominata "L'app che Protegge e Trasforma" finalizzata alla prevenzione e al supporto delle vittime di violenze di genere.

Il prodotto punta a stabilirsi come uno strumento intelligente e sicuro che aiuta l'utilizzatore a riconoscere segnali di pericolo e a fornire risorse per tutelarsi da atti, che siano fisici o psicologici, che potrebbero nuocere alla sua salute. L'applicazione implementa metodologie volte alla tutela delle persone che sono già vittima di violenze, ma anche per la valutazione e l'individuazione di situazioni che potrebbero sfociare in atti di violenza.

1.3. Glossario

I documenti prodotti nei processi di ciclo di vita sono corredati da un glossario che costituisce un riferimento condiviso. Il suo scopo è ridurre le ambiguità lessicali che possono emergere a causa dei diversi livelli di competenze tecniche tra i destinatari dei documenti. Poiché le Norme di Progetto, l'Analisi dei Requisiti, il Piano di Progetto e il Piano di Qualifica sono rivolti a figure eterogenee, tra cui acquirenti, fornitori, sviluppatori, gestori e utenti del prodotto software, è necessario che ogni termine potenzialmente ambiguo, tecnico o soggetto a interpretazioni multiple sia riportato in modo chiaro e univoco.

Per favorire la reperibilità dei termini e rendere immediata la loro identificazione all'interno dei documenti, ogni voce del glossario è richiamata nel testo mediante la seguente convenzione stabilita dal gruppo BitByBit:

termine_nel_glossario^G

Il glossario è considerato parte integrante del processo di documentazione: viene aggiornato ogni volta che emergono nuovi termini rilevanti, che vengono introdotti nuovi concetti tecnici o che si rende necessario chiarire ambiguità riscontrate durante la redazione o la verifica dei documenti.

1.4. Riferimenti

1.4.1 Riferimenti normativi

- Capitola d'appalto C4: L'app che Protegge e Trasforma
<https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C4.pdf>
- Standard ISO/IEC 12207:1995
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf

1.4.2 Riferimenti informativi

-

2. Processi primari

3. Processi di supporto

La presente sezione espone i seguenti processi di supporto:

- (a) **documentazione;**
- (b) **gestione della configurazione;**
- (c) **gestione della qualità;**
- (d) **verifica;**
- (e) **validazione.**

3.1. Documentazione

Il processo di documentazione ha l'obiettivo di definire le modalità, gli strumenti e le convenzioni che il gruppo adotta per la produzione, la gestione, la revisione e l'approvazione di tutti i documenti del progetto.

3.1.1 Obbiettivi

1. Garantire **completezza, chiarezza e coerenza** delle informazioni contenute nei documenti.
2. Assicurare la **tracciabilità** delle modifiche e delle versioni;
3. Favorire la **comprensibilità e manutenibilità** dei documenti da parte di tutti i membri del gruppo;
4. Fornire un **mezzo ufficiale di comunicazione** tra il gruppo, il committente e i revisori;
5. Rispettare la **uniformità formale e stilistica** tra i vari documenti prodotti.

L'implementazione del repository dedicato alla documentazione integra tutti gli strumenti necessari alla produzione dei documenti di ciclo di vita, garantendo che ogni nuovo membro del progetto possa svolgere le attività previste senza dipendere dal proprio ambiente di lavoro locale o incorrere in limitazioni tecniche.

È necessaria l'automazione dei processi dove è possibile, in modo da avere un sistema con pipeline per lo sviluppo che aiuti il membro del gruppo nei suoi compiti.

3.1.2 Strumenti a supporto

Per sostenere il processo di documentazione sono utilizzati i seguenti strumenti:

- **L^AT_EX**, per la stesura di documenti il gruppo ha scelto L^AT_EX, linguaggio di markup che permette di avere il controllo di tutte le parti di un documento, può essere eseguito nel repository senza la necessità di un editor di testo;
- **overleaf**, piattaforma online utilizzata in fase iniziale per l'apprendimento della sintassi e i comandi di L^AT_EX;
- **Google Sheets**: utilizzato per la redazione di pianificazioni, tracciamento di requisiti preliminari, matrici decisionali e supporto alla raccolta dati;
- **Google Docs**, impiegato durante riunioni o fasi preliminari per appunti condivisi e raccolta rapida di contenuti;
- **Google Presentazioni**: utilizzato per la redazione e condivisione dei diari di bordo richiesti dal corso;
- **GitHub**, piattaforma di versionamento distribuito e piattaforma di collaborazione utilizzata per la gestione centralizzata dei documenti, delle modifiche, dei branch e delle pull request;
- **GitHub Actions**, strumento di automazione per l'aggiunta del template comune a tutti i documenti e per la compilazione dei file `.tex` in PDF.

3.1.3 Tipologie di documenti

- **Documenti interni**, destinati alla gestione e al coordinamento interno (verbali, norme di progetto, glossario);
- **Documenti esterni**, destinati a revisori, docenti o aziende (Analisi dei Requisiti, Piano di Progetto, Piano di Qualifica);
- **Verbali interni ed esterni**, registrano rispettivamente, gli incontri tra i membri del gruppo e gli incontri tra il gruppo e gli enti esterni come l'azienda proponente o i professori.

Ogni documento, indipendentemente dalla sua categoria, deve essere redatto in conformità alle convenzioni formali definite nelle Norme di Progetto.

3.1.4 Implementazione di processo

Ogni documento, prima di essere integrato nella versione stabile del **repository^G**, deve attraversare un ciclo strutturato di produzione che garantisce la qualità, la verificabilità e la tracciabilità delle modifiche. Le attività descritte in questa sezione definiscono il flusso operativo seguito dal gruppo per l'elaborazione, la revisione e la pubblicazione dei prodotti documentali.

3.1.4.1 Creazione delle issue GitHub Ogni modifica documentale, compresa anche la creazione di un documento, inizia con l'apertura di una issue su GitHub per permettere la tracciabilità nella board del **Project GitHub^G**. Ogni commit deve essere atomico, cioè riferirsi ad un'unica azione in quanto migliorano la revisione, il recupero e la collaborazione.

3.1.4.2 Assegnazione della issue Nessuna issue può essere avviata senza una chiara assegnazione. L'assegnazione viene effettuata dal **responsabile^G**, che determina chi, nel gruppo, è incaricato dell'esecuzione dell'attività. Questo passaggio garantisce che ogni modifica sia riconducibile a un autore e che la responsabilità operativa sia chiaramente definita.

3.1.4.3 Impostazione del documento Il membro del team può ora avviare la stesura del documento all'interno del branch dedicato, il cui nome segue la nomenclatura presente nella sezione (da aggiungere). La prima attività consiste nella creazione del file sorgente, denominato secondo le convenzioni stabilite (ad esempio nel formato `nome_documento.tex`). Una volta inserito il nuovo file nel repository remoto, il sistema provvede automaticamente ad applicare il template comune, garantendo l'integrazione delle componenti strutturali condivise.

Spetta quindi al redattore adattare il template al contenuto specifico del documento, completandone le sezioni pertinenti e verificando che l'impostazione risultante sia conforme agli standard definiti per la documentazione del progetto. Questa attività assicura uniformità, leggibilità e coerenza stilistica dell'intera produzione documentale lungo tutto il ciclo di vita del progetto.

Ogni membro del gruppo è quindi tenuto a verificare che i documenti prodotti rispettino un formato uniforme, verificabile e riconducibile agli standard scelti.

Questa attività garantisce la leggibilità, la coerenza e la mantenibilità dell'intera produzione documentale durante tutto il ciclo di vita del progetto.

Intestazione

Ogni pagina del documento include un'intestazione che riporta il titolo dello specifico documento e il nome del gruppo.

Frontespizio

Il frontespizio possiede i seguenti elementi:

- logo dell'ateneo;
- informazioni dell'ateneo;
- logo del gruppo;
- contatto ufficiale del gruppo;
- nome del documento.

Storico del documento

Nell'ultima pagina deve essere presente uno storico delle modifiche apportate al documento. Le entry della tabella devono essere in ordine cronologico in modo che la prima entry faccia riferimento alla modifica più recente. Una entry è composta da:

1. **versione**;
2. **data della modifica**;
3. **autore della modifica**;
4. **verificatore**, il nome del verificatore che ha verificato quella modifica.

Versione	Data	Autore	Descrizione	Verificatore

Indice

Ogni documento deve presentare un indice delle sezioni e sottosezioni per una visita veloce e facilitata di specifici all'interno dello stesso documento.

Struttura dei verbali

I verbali rappresentano i rendiconti di riunioni ufficiali interne, a cui partecipano solo i componenti del gruppo, ed esterne, a cui invece partecipano anche enti esterni al gruppo come le aziende fornitrici o i professori. Ogni verbale oltre alle sezioni precedentemente elencate deve presentare al suo interno i seguenti elementi nell'ordine in cui sono descritti:

1. **Informazioni generali:** contiene i dati identificativi dell'incontro quali:

- il redattore;
- il verificatore;
- il responsabile;
- l'amministratore;
- la data in cui si è svolto l'incontro;
- la durata;
- l'elenco dei partecipanti e degli assenti.

Queste informazioni sono volte a dare un contesto formale alla riunione.

2. **Ordine del giorno:** riassume i punti principali che si intendono discutere durante l'incontro, predisposti in anticipo dal responsabile o dall'amministratore. Questa sezione funge da guida per la conduzione del verbale.
3. **Discussioni:** riporta in modo sintetico ma chiaro i contenuti emersi durante il confronto tra i partecipanti. Devono essere descritte le osservazioni, le problematiche sollevate e le proposte di soluzione valutate durante l'incontro.
4. **Decisioni:** elenca tutte le decisioni approvate dal gruppo, incluse quelle riguardanti modifiche ai documenti, scadenze o variazioni organizzative. Ogni decisione deve essere formulata in modo oggettivo e deve avere un codice identificativo univoco che segue il seguente formato:

[VI/VE] [fase a cui appartiene] Y.Z

dove:

- **VI/VE**, indica se la decisione fa riferimento ad un verbale interno o esterno, seguendo le convenzioni di scrittura stabilite.

- **fase**, indica la fase di progetto a cui si riferisce;
 - **Y**, indica il numero del verbale in riferimento a quella revisione;
 - **X** indica il numero della decisione interna a quel documento.
5. **To-do**: presenta, sotto forma di tabella, le attività operative pianificate a seguito della riunione. Ogni attività è identificata da:
- una descrizione sintetica;
 - Il codice della decisione da cui è stata creata (per ogni decisione ci possono essere 0 o più attività);
 - il numero della issue GitHub di quella attività presente nel project.

Convenzioni di scrittura

Le convenzioni di scrittura stabiliscono regole comuni per la redazione di tutti i documenti del progetto.

3.1.4.4 Date

- Le **date** devono essere nel formato **AAAA-MM-GG**.

Le ore hanno 2 formati:

- **Durata**: Per esprimere una durata usare il formato **Hh Mm** (es: 2h 15m)
- **Ora di orologio**: Per indicare che un evento è avvenuto in una specifica ora, usare il formato **HH:MM** (es: 15:45)

Nomi dei file

- Tutti i file devono avere nomi in minuscolo, senza spazi, e le parole devono essere separate da un **underscore** (**_**).
- Il nome di ogni verbale deve iniziare con **verbale_**, segue il tipo di verbale **interno** o **esterno** e la data in cui si è svolto. Un esempio di nominazione corretta è **verbale_interno_2025-11-05**.
- L'unica eccezione è nel caso in cui usare questo formato generi due verbali con lo stesso nome all'interno della stessa cartella. In tal caso è necessario aggiungere, dopo la data, una breve descrizione del contesto relativo al verbale, sostituendo agli spazi il simbolo **underscore** (**_**). Ad esempio **verbale_2025-10-28_incontro_c8**.
- I file PDF mantengono lo stesso nome del file sorgente **.tex**.

Sigle

Di seguito sono presenti le sigle usate all'interno dei documenti.

- **Fasi di progetto:**
 - **C**, Candidatura;
 - **RTB**, Requirements and Tecnology Baseline;
 - **PB**, Product Baseline.
- **Tipologie di documenti:**
 - **AdR**, analisi dei requisiti;
 - **NdP**, norme di progetto;
 - **PdQ**, piano di qualifica;
 - **PdP**, piano di progetto;
 - **G**, glossario;
 - **VI**, verbale interno;
 - **VE**, verbale esterno.

Nomi dei membri

- I nomi dei membri devono essere riportati nel formato: **Nome Cognome** (iniziali maiuscole, nessuna abbreviazione).

3.1.4.5 Modifiche a documenti Per eseguire delle modifiche alla documentazione di progetto, il membro del gruppo è tenuto a istanziare un nuovo branch di **feature**. La denominazione del branch di feature deve seguire la nomenclatura presente nella sezione (da aggiungere).

Ogni modifica a un documento deve essere accompagnata da un aggiornamento del numero di versione appropriato. Il membro del gruppo che si occupa delle modifiche è anche tenuto ad aggiornare lo **storico dei documenti** con le informazioni richieste.

Ogni modifica ad un documento deve essere tracciabile all'interno dello stesso tramite lo **storico del documento**, riportando:

- il numero di versione assegnato;
- la data della conclusione della modifica;
- il nome e cognome del membro del team che l'ha eseguita;
- una descrizione concisa, sintetica e formale delle modifiche avvenute;
- il nome del verificatore che è tenuto ad eseguire la verifica.

3.1.4.6 Conclusione modifiche Una volta completate le modifiche nel branch di **feature**, il membro incaricato procede ad aprire una **pull request^G** dal proprio branch di **feature** verso il branch **develop**. Nella fase di apertura della **pull request^G** devono essere compilate le informazioni richieste dal relativo template, specifico per il **repository^G** della documentazione. La **pull request^G** costituisce la richiesta formale per integrare le modifiche nella versione di sviluppo, permettendo al team di effettuare la revisione del codice e di approvarne l'unione.

3.1.4.7 Verifica Il verificatore esamina sia la correttezza formale sia sostanziale del documento.

- Se le modifiche risultano corrette, il verificato approva la **pull request^G**.
- Se emergono incongruenze o problemi, il verificatore richiede modifiche aggiungendo dei commenti specifici al codice nel messaggio della **pull request^G** e aprendo una issue **GitHub^G** dedicata.

Le modifiche richieste devono essere applicate dall'autore originario della **pull request^G**, che procede aggiornando il branch **develop** fino alla completa risoluzione. I verificatori sono tenuti a controllare che i documenti di ciclo di vita del progetto comprendano gli elementi di struttura elencati di seguito nell'ordine in cui si presentano.

3.1.4.8 Approvazione e pull request Una volta ottenuta l'approvazione del verificatore, la **pull request^G** viene sottoposta al **responsabile^G**, che effettua un'ultima revisione formale. Solo il **responsabile^G** è autorizzato a procedere con il merge della PR nel branch **develop**. Il merge rappresenta l'integrazione definitiva del documento nella versione stabile del progetto.

3.1.4.9 Pubblicazione Con la conclusione del merge, il documento aggiornato diventa parte integrante della **baseline^G** di progetto. Il branch **main** costituisce, infatti, il riferimento ufficiale per la versione stabile e coerente dei documenti.

3.2. Processo di gestione della configurazione

4. Processi organizzativi del ciclo di vita

4.1. Gestione dei processi