



**Università degli Studi di Padova**

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2021/2022



Gruppo: MERL

Email: merlunipd@gmail.com

# Norme di Progetto

# Registro delle Modifiche

Versione	Data	Autore/Verificatore	Modifica
v0.0.7	18/12/2021	Emanuele Pase Marco Mazzucato	Aggiunta sezione "Sviluppo"
v0.0.6	14/12/2021	Marco Mamprin Riccardo Contin	Aggiunta sezione "Formazione"
v0.0.5	14/12/2021	Marco Mamprin Riccardo Contin	Aggiunta sezione "Validazione"
v0.0.4	14/12/2021	Marco Mamprin Riccardo Contin	Aggiunta sezione "Gestione della Qualità"
v0.0.3	11/12/2021	Lorenzo Onelia Emanuele Pase	Aggiunta sezione "Verifica"
v0.0.2	10/12/2021	Emanuele Pase Riccardo Contin	Aggiunta sezione "Gestione della configurazione"
v0.0.1	08/12/2021	Marko Vukovic Marco Mazzucato	Aggiunta sezione "Infrastruttura"
v0.0.0	07/12/2021	Marko Vukovic Marco Mazzucato	Creata prima struttura del documento

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Processi Primari</b>	<b>6</b>
2.1	Fornitura . . . . .	6
2.2	Sviluppo . . . . .	6
2.2.1	Descrizione . . . . .	6
2.2.2	Attività . . . . .	6
2.2.3	Analisi dei Requisiti . . . . .	6
2.2.4	Progettazione . . . . .	8
2.2.5	Codifica . . . . .	8
<b>3</b>	<b>Processi di Supporto</b>	<b>10</b>
3.1	Documentazione . . . . .	10
3.2	Gestione della Configurazione . . . . .	10
3.2.1	Scopo . . . . .	10
3.2.2	Aspettative . . . . .	10
3.2.3	Descrizione . . . . .	10
3.2.4	Versionamento . . . . .	10
3.2.5	Sistemi software utilizzati . . . . .	11
3.2.6	Struttura repository . . . . .	11
3.3	Gestione della Qualità . . . . .	12
3.3.1	Scopo . . . . .	12
3.3.2	Piano di Qualifica . . . . .	12
3.3.3	Testing . . . . .	12
3.3.4	Aspettative . . . . .	13
3.4	Verifica . . . . .	13
3.4.1	Scopo . . . . .	13
3.4.2	Descrizione . . . . .	13
3.4.3	Indicazioni . . . . .	13
3.4.4	Verifica della documentazione . . . . .	14
3.5	Validazione . . . . .	14

3.5.1	Descrizione . . . . .	14
<b>4</b>	<b>Processi Organizzativi</b>	<b>15</b>
4.1	Gestione di Processo . . . . .	15
4.2	Infrastruttura . . . . .	15
4.2.1	Strumenti . . . . .	15
4.3	Miglioramento . . . . .	19
4.4	Formazione . . . . .	19

# 1. Introduzione

## 2. Processi Primari

### 2.1 Fornitura

### 2.2 Sviluppo

#### 2.2.1 Descrizione

Lo scopo del processo di sviluppo è quello di raggruppare compiti e attività relative alla codifica di un prodotto software, applicandole durante il suo intero ciclo di vita.

Al fine di produrre un software che rispetti le aspettative del proponente è necessario:

- Determinare i vincoli tecnologici;
- Determinare gli obiettivi di sviluppo e design;
- Realizzare un prodotto software che superi tutti i test di verifica e di validazione.

#### 2.2.2 Attività

Il processo di sviluppo prevede l'esecuzione delle seguenti attività:

- **Analisi dei Requisiti**
- **Progettazione**
- **Codifica**

#### 2.2.3 Analisi dei Requisiti

L'*Analisi dei Requisiti* è quell'attività che precede lo sviluppo e ha la funzione:

- Di definire lo scopo del prodotto da realizzare;
- Definire gli attori del sistema;
- Fissare le funzionalità del prodotto;
- Fornire una visione più chiara del problema ai progettisti;
- Fornire un riferimento ai verificatori per l'attività di controllo dei test;
- Fornire una stima della mole di lavoro.

### **Scopo**

Lo scopo dell'attività è quello di fornire in un documento tutti i requisiti individuati.

Al fine di individuarli è necessario:

- Leggere e comprendere la specifica del capitolato;
- Mantenere un confronto costante con il proponente.

### **Casi d'uso**

Definiscono uno scenario in cui uno o più attori interagiscono con il sistema. Sono identificati nel modo seguente:

**UC[Numero caso d'uso][Sottocaso]-[Titolo caso d'uso]**

### **Struttura dei requisiti**

Il codice identificativo di ciascun requisito è di seguito riportato:

**R[Importanza][Tipologia][Codice]**

con:

- Importanza:
  - **1** requisito obbligatorio;
  - **2** requisito desiderabile ma non obbligatorio;
  - **3** requisito opzionale.
- Tipologia:
  - **V** Vincolo;

- **P** Prestazionale;
- **Q** Qualitativo;
- **F** Funzionale.
- Codice:  
Identificativo univoco del requisito.

## 2.2.4 Progettazione

### Scopo

Lo scopo della progettazione è quello di definire una possibile soluzione ai requisiti evidenziati dall'analisi.

### Descrizione

La progettazione è formata da due parti:

- **Progettazione logica:** motiva le tecnologie, i framework e le librerie usate per la realizzazione di un prodotto, dimostrandone l'adeguatezza nel PoC.  
Contiene:
  - I framework e le tecnologie utilizzate;
  - Il Proof of Concept;
  - I diagrammi UML.
- **Progettazione di dettaglio:** illustra la base architettuale del prodotto coerentemente a ciò che è previsto nella Progettazione logica.  
Contiene:
  - Diagrammi delle classi;
  - Tracciamento delle classi;
  - Test di unità per ogni componente.

## 2.2.5 Codifica

### Scopo

Lo scopo della codifica è quello di implementare le specifiche individuate in un prodotto utilizzabile.



## **Commenti**

Nel caso sorga la necessità di scrivere qualche commento al codice è preferibile che esso sia chiaro e conciso.

## **Nomi dei files**

I nomi dei files devono:

- Essere univoci;
- Esplicitare il contenuto dei files stessi.

## **3. Processi di Supporto**

### **3.1 Documentazione**

### **3.2 Gestione della Configurazione**

#### **3.2.1 Scopo**

Lo scopo di questa sezione è quello di esplicitare come il gruppo intende gestire la produzione di risorse durante l'intero sviluppo del progetto.

#### **3.2.2 Aspettative**

Le attese che il gruppo MERL si pone sono:

- Semplificare l'individuazione di conflitti ed errori
- Uniformare gli strumenti utilizzati
- Tracciare le modifiche e avere sempre a disposizione una versione precedente per ogni file

#### **3.2.3 Descrizione**

Il processo di gestione della configurazione ha lo scopo di creare un ambiente in cui la produzione di documentazione e di codice avviene in maniera sistematica, ordinata e standardizzata. Ciò è permesso raggruppando ed organizzando tutti gli strumenti e le regole adoperati.

#### **3.2.4 Versionamento**

##### **Codice di versione**

La storia di una risorsa deve sempre poter essere ricostruibile in quanto nel suo arco di vita essa subisce svariate modifiche. A tal scopo è fondamentale

l'introduzione di un sistema di identificazione della versione:

$$[X].[Y].[Z]$$

Dove:

- **X** indica la versione stabile, quella approvata dal responsabile del documento
- **Y** indica la versione controllata, quella controllata dal verificatore del documento
- **Z** indica la versione modificata, quella sottoposta a modifica da parte di un redattore

### 3.2.5 Sistemi software utilizzati

Per la gestione delle differenti versioni è stato deciso di utilizzare il sistema di versionamento distribuito Git; utilizzando il servizio GitHub per gestire la repository.

Per la suddivisione e la gestione dei lavori da svolgere il nostro gruppo ha deciso di appoggiarsi al servizio web Trello.

### 3.2.6 Struttura repository

#### Repository utilizzata

Il repository in cui tutti i documenti vengono caricati è pubblico e si trova al seguente indirizzo: <https://github.com/merlunipd/login-warrior>.

#### Organizzazione del repository

L'organizzazione del repository è riassunta di seguito:

- Ramo principale **Main** in cui è presente la documentazione, o le parti dei documenti, approvate da almeno un verificatore e dal responsabile
- Rami secondari utilizzati per redarre i vari paragrafi della documentazione che, una volta uniti al ramo principale attraverso l'attività di merging, vengono eliminati per evitare di ostruire il repository.

Nel ramo Main i documenti si trovano all'interno della cartella "documenti", all'interno della quale possiamo trovare:

- La cartella "candidatura" che contiene i documenti inerenti alla candidatura al capitolato C5
- La cartella "interni" che contiene i documenti utili al gruppo
- La cartella "esterni" contenente i documenti da fornire ai committenti e al proponente.

Sempre nel Main possiamo trovare il file ".gitignore" necessario ad evitare il tracciamento dei file  $\LaTeX$  non necessari ai fini del progetto.

## 3.3 Gestione della Qualità

### 3.3.1 Scopo

L'obiettivo della gestione della qualità è quello di garantire che i processi e il prodotto soddisfino le richieste del proponente e che lo facciano con la miglior qualità possibile.

Vogliamo, inoltre, poter ottenere un miglioramento continuo della nostra qualità, osservando il nostro andamento e tramite verifiche retrospettive.

### 3.3.2 Piano di Qualifica

Per coprire gli obiettivi di questo processo utilizziamo il Piano Di Qualifica, cioè un documento nel quale:

- Fissiamo gli obiettivi di qualità;
- Definiamo le metriche per avere una visione quantitativa;
- Definiamo dei test di qualità e funzionamento;
- Effettuiamo e documentiamo i test;
- Visualizziamo un cruscotto dello stato attuale;
- Discutiamo dei vari incontri con una verifica delle retrospettive;
- Proponiamo idee di auto-miglioramento.

### 3.3.3 Testing

**TODO:** Capire meglio i test da inserire

### 3.3.4 Aspettative

Con questo processo ci poniamo le seguenti aspettative:

- Assicurarci della qualità del prodotto che realizziamo;
- Assicurarci delle qualità dei processi e del gruppo;
- Poter avere una visione quantitativa del nostro avanzamento;
- Poter effettuare test frequentemente e che siano predicibili;
- Poterci migliorare progressivamente per non ripetere gli stessi errori;
- Soddisfare al meglio le aspettative del committente.

## 3.4 Verifica

### 3.4.1 Scopo

La verifica ha come obiettivo il controllo che un prodotto sia corretto e completo.

### 3.4.2 Descrizione

Nella verifica si prende in input ogni processo e lo si controlla, in modo da identificare dubbi o incorrettezze. Si applica la verifica su un processo quando:

- Si raggiunge un livello di maturità adeguato e sufficiente.
- A seguire di un cambiamento di stato.

### 3.4.3 Indicazioni

Per assicurare il corretto svolgimento della verifica si devono rispettare i seguenti punti:

- Seguire procedure definite.
- Seguire criteri validi e affidabili.
- Ogni prodotto deve passare attraverso fasi successive, che verranno quindi verificate.

Una volta terminata la fase di verifica, rispettando i punti appena citati si può quindi proseguire alla fase di **Validazione**.

### 3.4.4 Verifica della documentazione

Il processo di verifica per quanto riguarda la documentazione può essere riassunto in queste attività:

- Controllo dell'ortografia e della sintassi.
- Controllo della stesura del documento.
- Controllo della pertinenza e della correttezza dei contenuti del documento.

#### Attività di analisi statica

Questa tipologia di analisi risulta molto utile per la verifica di un documento. Si divide in:

- **Walkthrough:** Tecnica dove il verificatore va alla ricerca di eventuali errori attraverso una lettura ad ampio spettro.
- **Inspection:** Tecnica dove il verificatore va alla ricerca di specifici errori attraverso letture mirate.

Inizialmente l'attività di Walkthrough sarà prevalente rispetto a Inspection ma con l'avanzamento dell'attività di progetto e quindi con il ripetersi del processo di Verifica sulla documentazione, si riuscirà a sviluppare una lista degli errori più comuni detta **Lista di Controllo** consentendo l'utilizzo più massivo della tecnica **Inspection** che risulta essere più efficiente.

## 3.5 Validazione

### 3.5.1 Descrizione

Con Validazione si intende l'incontro con il committente in cui si presenta il proprio lavoro, in particolare il prodotto finale, ottenendo così l'approvazione o meno sull'operato.

Lo scopo è quello di accertarsi di aver conseguito i requisiti obbligatori minimi, imposti dal committente e in modo efficace ed efficiente. Per verificare il tutto si eseguirà un collaudo con il committente, che sarà composto da diversi test, che ne verifichino le qualità, e inoltre dovranno già essere eseguiti in precedenza ed essere predicibili, cioè senza risultati o comportamenti inattesi.

## 4. Processi Organizzativi

### 4.1 Gestione di Processo

### 4.2 Infrastruttura

Fanno parte dell'infrastruttura organizzativa tutti gli strumenti che permettono al gruppo di attuare in modo efficace ed efficiente i processi organizzativi. In particolare tali strumenti permettono la **comunicazione**, il **coordinamento** e la **pianificazione**.

#### 4.2.1 Strumenti

##### Trello

Principale strumento di project management utilizzato come **Kanban**.

La **board principale** è divisa nelle seguenti liste:

- **Backlog**: contiene task da fare in un momento non specificato. Questa lista viene esaminata nel momento di definizione di una milestone come integrazione per assicurarsi di non dimenticare obiettivi fondamentali;
- **Todo**: contiene task da fare il prima possibile. Se una task non è assegnata a nessuno, qualsiasi membro del gruppo può autonomamente prenderla in carico e portare avanti il suo ciclo di vita. L'unico che può inserire nuove task in questa lista è il **Responsabile di Progetto**;
- **Doing**: contiene task in corso. Da questo punto del ciclo di vita, qualsiasi task ha almeno un membro associato a essa, non possono esserci task "orfane". Questo permette di coordinarsi per essere sicuri che i membri del gruppo non stiano lavorando asincronamente a una stessa task;

- **Verify**: contiene task completate che necessitano di verifica. Qualsiasi membro del gruppo può autonomamente diventare verificatore di una task aggiungendosi a essa, con la precondizione che non sia la stessa persona che l'ha svolta. In caso di problemi può nascere una conversazione nella sezione "Commenti" tra l'esecutore e il verificatore, con possibile ritorno della task nella lista "Doing". Quando il verificatore conferma l'idoneità, può segnalare la sua approvazione con il corrispettivo tag "Approved". Sarà poi compito del **Responsabile di Progetto** accettare la fine della task e spostarla in "Done";
- **Done**: contiene task completate, verificate e accettate. Utile principalmente come traccia storica dei compiti svolti (chi, cosa, quando, problemi).

Ciascuna **task** è costituita da:

- **titolo** sintetico nella seguente forma generale: *[categoria] - [breve titolo significativo]* (e.g. `documentazione - verbale meeting interno 23/11/2021`);
- **descrizione** opzionale e breve per dettagli importanti;
- **membro/i** del gruppo coinvolti nella task;
- **tag** per segnalare particolarità della task. Le etichette principali definite sono:
  - **Importante**: segnala una task con alto livello di priorità;
  - **Team**: segnala una task che necessita del coinvolgimento dell'intero team. Rimane comunque che ci deve essere un membro incaricato responsabile per quella determinata task;
  - **Approved**: segnala una task in "Verify" che è stata verificata e può essere spostata (dal Responsabile di Progetto) in "Done".
- **commenti** per discutere (soprattutto asincronamente) sulla specifica task.

Le task inserite nel sistema non vengono mai cancellate ma solo spostate tra le liste presenti. Solo in casi eccezionali le task possono essere archiviate ma anche in questo caso di loro e della loro storia resta traccia.

Per navigare più facilmente nella bacheca è possibile impostare dei **filtri**, ad esempio per membro o tag.



## GitHub

È il principale servizio di hosting della repository di gruppo e di controllo della versione distribuito.

Generalmente il workflow adottato dal gruppo è il GitHub Flow, che sinteticamente segue il seguente schema:

- riallineamento della repository locale con quella remota;
- creazione di un branch locale su cui effettuare le modifiche;
- push del branch locale verso repository remota;
- creazione di una pull request;
- verifica e successivo merge del branch con le modifiche;
- eliminazione del branch utilizzato dalla repository remota.

Per i dettagli consultare la documentazione ufficiale:

- <https://docs.github.com/en/get-started/quickstart/github-flow>

GitHub offre un sistema di "Issue" e "Milestone" per pianificare e coordinare le attività da svolgere. Tipicamente sarà compito del **Responsabile di Progetto** predisporre le issue relative a una particolare milestone in modo che il resto del team possa avere più autonomia, potendo autoassegnarsi issue ancora aperte. Idealmente le issue saranno collegate con una o più pull request.

**TODO: il sistema di Issue e Milestone di GitHub deve ancora essere utilizzato e approfondito**

## Discord

Principale strumento di **comunicazione interna sincrona e asincrona**. Vengono utilizzati 3 categorie di canali:

- **Canali Risorse:** condivisione di risorse, creazione di sondaggi per effettuare decisione problematiche, integrazione con strumenti esterni per permettere notifiche (e.g. GitHub);
- **Canali Testuali:** comunicazioni testuali sincrone e asincrone tra i membri del gruppo;
- **Canali Vocali:** comunicazioni vocali tra i membri del gruppo, con possibilità di condividere lo schermo se necessario.

Ciascuna categoria può contenere un numero variabile di canali, a seconda delle necessità del periodo.

## WhatsApp

Principale strumento di **comunicazione interna testuale asincrona**. Viene utilizzato in due modalità:

- **Gruppo**: chat condivisa utilizzata, con parsimonia, per comunicazioni rivolte a tutti i membri;
- **Individuale**: ogni membro del gruppo può essere contattato singolarmente.

## Google Calendar

Calendario condiviso del gruppo utilizzato per comunicare e ricordare:

- **Meeting Interni**: di cui saranno specificati:
  - orario di inizio;
  - moderatore;
  - scribe (redazione verbale);
  - argomenti: specifiche idee da trattare durante una riunione del gruppo.
- **Meeting Esterni**: con proponente o committente;
- **Scadenze Interne**;
- **Scadenze Esterne**;
- qualsiasi altra attività o evento che può essere collocato in un tempo specifico.

Mantenere il calendario aggiornato è compito del Responsabile di Progetto.

## Google Drive

Strumento utilizzato come:

- **directory condivisa** dai membri del gruppo per documenti temporanei o non ufficiali;
- accesso alla **suite Google**: Docs, Sheets, Slides.

## Zoom

Strumento di videochiamata utilizzato principalmente per la comunicazione esterna con committente e proponente.

## Google Mail

Utilizzo dell'indirizzo e-mail condiviso `merlunipd@gmail.com` per le **comunicazione esterna** come gruppo con i proponenti e il committente.

## 4.3 Miglioramento

## 4.4 Formazione

Per garantire un andamento organizzato, simultaneo e alla pari, senza lasciare indietro nessuno, e che quindi favorisca un miglior lavoro asincrono, ogni componente del gruppo, in caso di lacune, dovrà studiarsi in autonomia gli strumenti e le tecnologie utilizzate per documentare e sviluppare il progetto, oppure condividere eventuali conoscenze con gli altri membri per velocizzare questo processo di formazione.

Di seguito sono riportati gli strumenti e le tecnologie utilizzate, con i principali riferimenti usati dal gruppo:

- LaTeX: [https://www.overleaf.com/learn/latex/Learn\\_LaTeX\\_in\\_30\\_minutes](https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes);
- Tabelle LaTeX: <https://www.tablesgenerator.com/>;
- GitHub: <https://docs.github.com>;
- **TODO: Tecnologie software.**