



---

# NORME DI PROGETTO

BLOCKCOVID

---

[dpcm2077@gmail.com](mailto:dpcm2077@gmail.com)

## INFORMAZIONI SUL DOCUMENTO

<b>Versione</b>	3.0.0
<b>Uso</b>	interno
<b>Stato</b>	approvato
<b>Destinatari</b>	Prof. Tullio Vardanega Prof. Riccardo Cardin DPCM 2077
<b>Redazione</b>	Antonio Badan
<b>Verifica</b>	Damiano Bertoldo
<b>Approvazione</b>	Samuele De Grandi

## Registro delle modifiche

Versione	Descrizione	Data	Autore	Ruolo
3.0.0	Approvazione documento	2021-03-24	Samuele De Grandi	Responsabile
2.0.1	Modifica e verifica §4.6, §2.1.5	2021-03-23	Antonio Badan Damiano Bertoldo	Amministratore Verificatore
2.0.0	Approvazione documento	2021-03-07	Damiano Bertoldo	Responsabile
1.1.5	Modifica e verifica §3.2.2.4	2021-03-07	Samuele De Grandi Matteo Budai	Amministratore Verificatore
1.1.4	Aggiunti e verificati IDE §4.5.4	2021-03-04	Antonio Badan Matteo Budai	Amministratore Verificatore
1.1.3	Approvazione di §3.1.4.3 §3.2.2.1 §3.3.3 §3.4.4.1 §4.2.1 §4.2.2 §4.2.3 §4.5.1 §4.5.2 §A	2021-02-17	Damiano Bertoldo	Responsabile
1.0.3	Modifica e verifica §4.2.1 §4.2.2 §4.2.3 §4.5.1 §4.5.2	2021-02-15	Ivan Piacere Matteo Budai	Amministratore Verificatore
1.0.2	Modifica e verifica §3.1.4.3 §3.2.2.1 §3.3.3 §3.4.4.1	2021-02-15	Ivan Piacere Matteo Budai	Amministratore Verificatore
1.0.1	Aggiunto §A del Piano di Qualifica	2021-02-15	Antonio Badan Matteo Budai	Amministratore Verificatore
1.0.0	Approvazione documento	2021-01-03	Ivan Piacere	Responsabile
0.0.5	Modifica e verifica §3.4.4.1	2021-01-03	Sara Privitera Antonio Badan	Amministratore Verificatore
0.0.4	Stesura e verifica §4	2021-01-02	Daniele Spigolon Antonio Badan	Amministratore Verificatore
0.0.3	Stesura e verifica §3	2021-01-02	Sara Privitera Antonio Badan	Amministratore Verificatore
0.0.2	Stesura e verifica §2	2021-01-01	Damiano Bertoldo Antonio Badan	Amministratore Verificatore

0.0.1	Stesura e verifica §1	2021-01-01	Daniele Spigolon Antonio Badan	Amministratore Verificatore
-------	-----------------------	------------	-----------------------------------	--------------------------------

## Indice

<b>1</b>	<b>Introduzione</b>	<b>7</b>
1.1	Scopo del Documento . . . . .	7
1.2	Composizione del Documento . . . . .	7
1.3	Scopo del prodotto . . . . .	7
1.4	Glossario e documenti . . . . .	7
1.5	Riferimenti . . . . .	8
1.5.1	Riferimenti Normativi . . . . .	8
1.5.2	Riferimenti Informativi . . . . .	8
<b>2</b>	<b>Processi Primari</b>	<b>9</b>
2.1	Fornitura . . . . .	9
2.1.1	Scopo . . . . .	9
2.1.2	Aspettative . . . . .	9
2.1.3	Descrizione . . . . .	10
2.1.4	Attività . . . . .	10
2.1.4.1	Studio di Fattibilità . . . . .	10
2.1.4.2	Piano di Progetto . . . . .	10
2.1.4.3	Piano di Qualifica . . . . .	11
2.1.5	Strumenti . . . . .	11
2.1.5.1	Fogli Google . . . . .	11
2.1.5.2	Gannt Project . . . . .	11
2.2	Sviluppo . . . . .	11
2.2.1	Scopo . . . . .	11
2.2.2	Aspettative . . . . .	11
2.2.3	Descrizione . . . . .	12
2.2.4	Attività . . . . .	12

2.2.4.1	Analisi dei requisiti . . . . .	12
2.2.4.2	Progettazione . . . . .	14
2.2.4.3	Codifica . . . . .	15
<b>3</b>	<b>Processi di supporto</b>	<b>17</b>
3.1	Documentazione . . . . .	17
3.1.1	Scopo . . . . .	17
3.1.2	Aspettative . . . . .	17
3.1.3	Descrizione . . . . .	17
3.1.4	Attività . . . . .	17
3.1.4.1	Implementazione del processo . . . . .	17
3.1.4.2	Sviluppo e design . . . . .	18
3.1.4.3	Norme tipografiche . . . . .	20
3.1.5	Strumenti . . . . .	22
3.1.5.1	LaTeX . . . . .	22
3.1.5.2	Google Drive . . . . .	22
3.1.5.3	Diagrammi . . . . .	22
3.2	Gestione della Configurazione . . . . .	23
3.2.1	Scopo . . . . .	23
3.2.2	Versionamento . . . . .	23
3.2.2.1	Codice di versione del documento . . . . .	23
3.2.2.2	Versionamento con GitHub . . . . .	23
3.2.2.3	Repository . . . . .	24
3.2.2.4	Configurazione del workflow . . . . .	24
3.3	Garanzia della Qualità . . . . .	24
3.3.1	Scopo . . . . .	24
3.3.2	Aspettative . . . . .	24
3.3.3	Descrizione . . . . .	25

3.3.4	Controllo qualità di prodotto . . . . .	25
3.3.5	Controllo qualità di processo . . . . .	25
3.3.6	Attività . . . . .	26
3.3.7	Strumenti . . . . .	26
3.4	Verifica . . . . .	26
3.4.1	Scopo . . . . .	26
3.4.2	Aspettative . . . . .	26
3.4.3	Descrizione . . . . .	26
3.4.4	Attività . . . . .	27
3.4.4.1	Analisi . . . . .	27
3.4.4.2	Test . . . . .	28
3.4.5	Strumenti . . . . .	30
3.4.5.1	Verifica ortografica . . . . .	30
3.5	Validazione . . . . .	31
3.5.1	Scopo . . . . .	31
3.5.2	Aspettative . . . . .	31
3.5.3	Descrizione . . . . .	31
3.5.4	Attività . . . . .	31
<b>4</b>	<b>Processi Organizzativi</b>	<b>32</b>
4.1	Gestione della comunicazione . . . . .	32
4.1.1	Comunicazioni interne . . . . .	32
4.1.2	Comunicazioni esterne . . . . .	32
4.1.2.1	Comunicazioni esterne scritte . . . . .	33
4.2	Riunioni . . . . .	33
4.2.1	Verbali di riunione . . . . .	33
4.2.2	Riunioni interne . . . . .	34
4.2.3	Riunioni esterne . . . . .	35

4.3	Ruoli di progetto . . . . .	36
4.3.1	Responsabile di progetto . . . . .	37
4.3.2	Amministratore di progetto . . . . .	37
4.3.3	Analista . . . . .	37
4.3.4	Progettista . . . . .	38
4.3.5	Programmatore . . . . .	38
4.3.6	Verificatore . . . . .	38
4.4	Formazione del personale . . . . .	39
4.5	Ambiente di lavoro . . . . .	39
4.5.1	Ticketing . . . . .	39
4.5.2	Coordinamento attività . . . . .	40
4.5.3	Documentazione . . . . .	41
4.5.4	Ambiente di sviluppo . . . . .	41
4.6	Strumenti organizzativi . . . . .	42
<b>A</b>	<b>Standard di Qualità</b>	<b>43</b>
A.1	ISO/IEC 9126 . . . . .	43
A.1.1	Metriche per la qualità interna . . . . .	43
A.1.2	Metriche per la qualità esterna . . . . .	43
A.1.3	Metriche per la qualità in uso . . . . .	43
A.1.4	Modello della qualità del software . . . . .	44
A.1.4.1	Funzionalità . . . . .	44
A.1.4.2	Affidabilità . . . . .	44
A.1.4.3	Efficienza . . . . .	45
A.1.4.4	Usabilità . . . . .	45
A.1.4.5	Manutenibilità . . . . .	45
A.1.4.6	Portabilità . . . . .	46

# 1 Introduzione

## 1.1 Scopo del Documento

Lo scopo di questo documento è fissare tutte le regole e le procedure fondamentali per assicurare al GRUPPO<sub>G</sub> DPCM 2077 un modo di lavorare comune che garantisca quantificabilità, efficacia ed efficienza. In quest'ottica, questo documento deve essere visionato da tutti i componenti di DPCM 2077 a ogni sua approvazione. I membri di DPCM 2077 sono obbligati ad applicare quanto scritto al fine di mantenere omogeneità e coesione in ogni aspetto del progetto. Saranno inoltre elencati e discussi tutti gli strumenti software scelti internamente al gruppo per rispettare le regole e attuare le procedure.

## 1.2 Composizione del Documento

La struttura del documento si ispira alla suddivisione dei PROCESSI<sub>G</sub> in tre tipologie desunta dallo standard ISO/IEC 12207:1995:

- **Primari:** acquisizione, fornitura, sviluppo, gestione operativa e manutenzione;<sup>1</sup>
- **Supporto:** documentazione, gestione della configurazione, accertamento della qualità, verifica, validazione, revisioni congiunte con il cliente, verifiche ispettive interne e risoluzione dei problemi;
- **Organizzativi:** gestione dei processi, gestione delle infrastrutture tecniche, miglioramento del processo e formazione del personale.

## 1.3 Scopo del prodotto

Il prodotto da sviluppare ha lo scopo di monitorare e regolare l'utilizzo e l'igienizzazione delle POSTAZIONI<sub>G</sub> all'interno di uno spazio condiviso, al fine di ridurre il rischio di trasmissione di un'infezione e di adempiere alle norme di legge.

## 1.4 Glossario e documenti

All'interno del documento sono presenti termini che assumono significati diversi a seconda del contesto. Per evitare ambiguità, è stato creato un documento di nome Glossario che conterrà tali termini con il loro significato specifico. Per segnalare che un termine del testo è presente all'interno del Glossario, verrà aggiunta una G pedice posta a fianco del termine ambiguo. Quando si fa riferimento a un altro documento riguardante questo progetto vi si pone a pedice una D.

---

<sup>1</sup>nel presente documento vengono trattati solamente *fornitura* e *sviluppo*



## 1.5 Riferimenti

### 1.5.1 Riferimenti Normativi

- C1  
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C1.pdf>
- VE\_2021\_01\_22 v. 1.0.0<sub>D</sub>

### 1.5.2 Riferimenti Informativi

- PIANO DI QUALIFICA v. 2.0.0<sub>D</sub>
- Standard ISO/IEC 12207:1995  
[https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf)
- Standard ISO 8601  
<https://www.iso.org/iso-8601-date-and-time-format.html>
- Sito ufficiale di GIT<sub>G</sub>  
<https://git-scm.com/>
- Documentazione ufficiale di Git  
<https://git-scm.com/doc>
- L'Arte di Scrivere con  $\LaTeX$   
[http://www.lorenzopantieri.net/LaTeX\\_files/ArteLaTeX.pdf](http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf)
- Libro di Ingegneria del Software: Software Engineering - Ian Sommerville - 10th edition
- Sito ufficiale di Python  
<https://www.python.org/>
- Sito ufficiale di Angular  
<https://angular.io/>
- Sito ufficiale di JavaScript  
<https://www.javascript.com/>
- Sito ufficiale di TypeScript  
<https://www.typescriptlang.org/>
- Sito ufficiale di Solidity  
<https://docs.soliditylang.org/en/v0.8.2/>
- Sito ufficiale di Django  
<https://www.djangoproject.com/>
- Sito ufficiale di Truffle  
<https://www.trufflesuite.com/>
- Sito ufficiale di Geth  
<https://geth.ethereum.org/downloads/>

## 2 Processi Primari

### 2.1 Fornitura

#### 2.1.1 Scopo

Lo scopo del processo di fornitura è di determinare le procedure e le risorse necessarie allo svolgimento del PROGETTO<sub>G</sub>. È necessario che il FORNITORE<sub>G</sub> esegua un'attività di analisi e successiva redazione dello Studio di Fattibilità: un documento che comprende le richieste del PROPONENTE<sub>G</sub>, individua i potenziali rischi e criticità del C1. È inoltre necessario stipulare e concordare con il proponente un contratto per la consegna e manutenzione del prodotto. Si passa dunque alla determinazione delle procedure e risorse necessarie, che andranno a confluire nel Piano di Progetto che porrà le basi per la consegna e realizzazione del prodotto.

Il PROCESSO<sub>G</sub> di fornitura è composto dalle seguenti attività:

- avvio;
- approfondimento di risposte alle richieste;
- contrattazione;
- pianificazione;
- esecuzione e controllo;
- revisione e valutazione;
- consegna e completamento.

#### 2.1.2 Aspettative

Durante lo svolgimento dell'intero progetto, DPCM 2077 intende mantenere un costante dialogo con il proponente, per avere un riscontro efficace sul lavoro svolto.

Tale dialogo permette di arrivare a una collaborazione per quanto riguarda:

- determinare gli aspetti chiave che il prodotto dovrà soddisfare;
- stilare requisiti e vincoli sui processi;
- stimare le tempistiche di lavoro;
- chiarire eventuali dubbi emersi;
- stimare i costi;
- accordarsi sulla qualifica del prodotto.

### 2.1.3 Descrizione

Questo documento tratta le norme a cui i membri del gruppo DPCM 2077 devono attenersi per diventare fornitori di Imola Informatica S.p.A e dei committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin. Ogni membro è tenuto ad attenersi a quanto esposto durante la progettazione, lo sviluppo e la consegna del prodotto.

### 2.1.4 Attività

#### 2.1.4.1 Studio di Fattibilità

In seguito alla presentazione ufficiale dei CAPITOLATI D'APPALTO<sub>G</sub>, il Responsabile di Progetto organizza riunioni con i membri del gruppo DPCM 2077 al fine di permettere lo scambio di opinioni sui capitolati proposti. Lo Studio di Fattibilità, redatto dall'Analista, indica per ogni capitolato:

- **Informazioni generali:** vengono elencate le informazioni generali del capitolato, quali nome del progetto, proponente e committente;
- **Descrizione del capitolato:** è una sintesi su cosa viene richiesto per lo sviluppo del capitolato;
- **Studio del dominio:** elenco di quali tecnologie, o applicazioni, sono richieste, o consigliate, per lo sviluppo;
- **Pro e Contro:** considerazioni del gruppo su fattori positivi o criticità del capitolato;
- **Conclusioni:** vengono espone le ragione per il quale il gruppo ha accettato o rifiutato il capitolato.

#### 2.1.4.2 Piano di Progetto

Il RESPONSABILE<sub>G</sub>, che ne ha in capo la redazione, redige un Piano di Progetto da seguire durante il corso del progetto.

Il documento è composto in questo modo:

- **Analisi dei rischi:** vengono analizzati i rischi che potranno presentarsi. Viene anche analizzato la frequenza con cui potrà presentarsi il rischio con la relativa gravità;
- **Modello di sviluppo:** viene descritto il modello di sviluppo;
- **Pianificazione:** vengono pianificate le attività da eseguire durante il progetto e le rispettive scadenze;
- **Preventivo:** viene data una stima del lavoro necessario per ciascuna fase proponendo un preventivo per il costo totale;
- **Consuntivo:** viene tracciato un consuntivo di periodo relativo all'andamento rispetto a ciò che è stato preventivato.

### 2.1.4.3 Piano di Qualifica

I VERIFICATORI<sub>G</sub> dovranno occuparsi del documento Piano di Qualifica, che contiene le linee guida da adottare per garantire la qualità del materiale prodotto dal gruppo.

Il piano è così suddiviso:

- **Qualità di processo:** vengono identificati gli standard dei processi, stabiliti degli obiettivi, escogitate delle strategie per attuarli e individuate le metriche per misurarli e controllarli;
- **Qualità di prodotto:** vengono identificati gli attributi più rilevanti per il prodotto, definiti degli obiettivi per raggiungerli e delle metriche per misurarli;
- **Specifiche dei test:** definiscono una serie di test attraverso i quali il prodotto passa per garantire che soddisfi i requisiti;
- **Standard di qualità:** vengono esposti gli standard di qualità scelti;
- **Valutazioni per il miglioramento:** vengono riportati i problemi e le relative soluzioni nel ricoprire un determinato ruolo e nell'uso degli strumenti scelti;
- **Resoconto delle attività di verifica:** per ogni attività si riportano i risultati delle metriche calcolate in forma di resoconto.

### 2.1.5 Strumenti

#### 2.1.5.1 Fogli Google

Software per la creazione di fogli elettronici utilizzato all'interno del Drive del gruppo. Usato per fare calcoli, produrre diagrammi, istogrammi e creare tabelle.

#### 2.1.5.2 Gantt Project

Utilizzato dai responsabili di progetto nella pianificazione, nell'assegnazione delle risorse, nella verifica del rispetto dei tempi attraverso la creazione di diagrammi di Gantt.

## 2.2 Sviluppo

### 2.2.1 Scopo

Il processo di sviluppo contiene le attività e i compiti da svolgere, relative al prodotto da sviluppare.

### 2.2.2 Aspettative

Le aspettative, riguardo il processo in questione, sono le seguenti:

- stabilire gli obiettivi di sviluppo;
- stabilire i vincoli tecnologici;
- stabilire i vincoli di design;
- realizzare un prodotto finale che superi i test, che soddisfi i requisiti e le richieste del proponente.

### 2.2.3 Descrizione

Il processo di sviluppo, secondo lo standard ISO/IEC 12207:1995, si articola in:

- Analisi dei requisiti;
- Progettazione;
- Codifica.

### 2.2.4 Attività

Di seguito verranno analizzate le varie sezioni elencate precedentemente in modo dettagliato.

#### 2.2.4.1 Analisi dei requisiti

##### Scopo

Gli analisti hanno in capo la stesura dell'Analisi dei Requisiti, il cui scopo è elencare e definire tutti i requisiti del C1.

Il documento conterrà:

- descrizione generale del prodotto;
- fornire ai PROGETTISTI<sub>G</sub> riferimenti precisi ed affidabili;
- funzionalità e requisiti concordati con il cliente;
- casi d'uso rappresentati tramite diagrammi UML<sub>G</sub>;
- calcolo della mole di lavoro per poter eseguire una stima dei costi.

##### Aspettative

L'obiettivo dell'attività di analisi consiste nella creazione della documentazione formale contenente tutti i requisiti richiesti dal proponente.

## Requisiti

Al fine di facilitarne la lettura e comprensione, ogni requisito viene classificato univocamente come segue:

**R[Priorità][Tipologia][Codice]**

Dove:

- **R**: abbreviazione di requisito;
- **Priorità**: ogni requisito assumerà uno dei seguenti valori:
  - **1**: requisito obbligatorio, ossia irrinunciabile;
  - **2**: requisito desiderabile, non strettamente necessario ma a valore aggiunto;
  - **3**: requisito opzionale, ossia relativamente utile o contrattabile in corso d'opera.
- **Tipologia** : ogni requisito assumerà uno dei seguenti valori:
  - **F**: funzionale;
  - **P**: prestazionale;
  - **Q**: qualitativo;
  - **V**: vincolo.
- **Codice**: codice identificativo univoco in forma gerarchica padre/figlio rappresentato come segue:

**[padre].[figlio]**

## Casi d'uso

La convenzione scelta per la rappresentazione univoca dei casi d'uso è la seguente:

**UC[codicePadre].[codiceFiglio]**

Dove:

- **UC**: acronimo utilizzato per *Use Case*;
- **codicePadre**: identificativo univoco del caso d'uso;
- **codiceFiglio**: identificativo dei sottocasi, può contenere altri sottolivelli.

Ogni caso d'uso deve contenere, integralmente o parzialmente, i seguenti campi:

- diagramma UML;

- attori primari;
- attori secondari;
- scenario principale;
- scopo e descrizione;
- estensioni (se presenti);
- inclusioni (se presenti);
- preconditione;
- postcondizione.

## UML

I diagrammi UML dovranno essere realizzati con la versione 2.0 del linguaggio.

### 2.2.4.2 Progettazione

#### Scopo

L'attività di progettazione individua, una volta concluso il documento Analisi dei Requisiti, le caratteristiche del prodotto richiesto e la soluzione migliore adottabile che soddisfi gli  $STAKEHOLDERS_G$ .

#### Aspettative

Realizzazione dell'architettura del sistema.

#### Requisiti

Si divide in:

- **Technology baseline:** contiene le specifiche della progettazione ad alto livello del prodotto e delle sue componenti, l'elenco dei diagrammi UML che saranno utilizzati per la realizzazione dell'architettura e i test di verifica;
- **Product baseline:** specifica ulteriormente l'attività di progettazione, integrando ciò che è riportato nella *Technology baseline*. Definisce inoltre i test necessari alla verifica;
- **Diagrammi UML:**
  - diagrammi delle classi;
  - diagrammi dei package;
  - diagrammi delle attività;
  - diagrammi di sequenza.
- **Tecnologie utilizzate:** descrivono le tecnologie utilizzate specificandone il loro utilizzo all'interno del progetto.

### 2.2.4.3 Codifica

#### Scopo

La codifica, svolta dal PROGRAMMATORE<sub>G</sub>, ha come scopo quello di normare l'effettiva realizzazione del prodotto software richiesto. I PROGRAMMATORI<sub>G</sub> dovranno attenersi a queste norme durante la programmazione e l'implementazione.

#### Aspettative

L'obiettivo dell'attività è la creazione di un prodotto software che rispetti le richieste stabilite con il proponente. L'uso di norme e convenzioni in questa fase è fondamentale per permettere la generazione di codice leggibile ed uniforme, agevolare la manutenzione e verificare, validare e migliorare la qualità del prodotto.

#### Descrizione

La scrittura del codice deve rispettare quanto stabilito nella documentazione di prodotto. Dovrà perseguire gli obiettivi descritti all'interno del documento PIANO DI QUALIFICA v2.0.0<sub>D</sub> al fine di garantire la scrittura di codice di qualità.

#### Stile di codifica

Al fine di garantire uniformità nella generazione del codice del progetto, ciascun membro del gruppo è tenuto a rispettare le seguenti norme:

- **Indentazione:** i blocchi innestati devono seguire una corretta indentazione, usando per ciascun livello di indentazione quattro (4) spazi (fatta eccezione per i commenti). Al fine di assicurare il rispetto di questa regola si consiglia di configurare adeguatamente il proprio IDE<sub>G</sub>;
- **Parentesi:** inserimento delle parentesi di delimitazione dei costrutti in linea e non al di sotto di essi;
- **Struttura dei metodi:** è desiderabile, ove possibile, avere i metodi strutturati in questo modo:
  - il nome deve iniziare con una lettera minuscola. Se il metodo è composto da più parole, quelle successive devono iniziare con la lettera maiuscola;
  - lunghezza del metodo breve;
  - la parentesi graffa di apertura deve essere nella stessa riga del metodo, separata da una singola spaziatura dopo il nome del metodo.
- **Univocità dei nomi:** classi, metodi, variabili devono avere un nome univoco ed esplicativo al fine di evitare ambiguità ed incomprensioni;
- **Classi:** i nomi delle classi devono iniziare con la lettera maiuscola. Se sono composte da più parole, le successive avranno la lettera maiuscola;
- **Costanti:** i nomi delle costanti devono essere scritte totalmente in maiuscolo. Se sono composte da più parole, saranno separate dal carattere *underscore* (`_`);



- **Lingua:** i nomi delle variabili, metodi, classi, costruttori e commenti saranno scritte in inglese.

### **Ricorsione**

L'uso della ricorsione va evitato quanto più possibile in quanto potrebbe indurre a una maggiore occupazione di memoria rispetto a soluzioni iterative.

## 3 Processi di supporto

### 3.1 Documentazione

#### 3.1.1 Scopo

Lo scopo di questa sezione è definire gli standard e le regole che riguardano la stesura e l'approvazione dei documenti. I documenti sono consultabili nelle apposite sezioni della REPOSITORY<sub>G</sub>:

<https://github.com/DPCMGroup/dpcm2077-docs>

#### 3.1.2 Aspettative

Si vuole fornire uno strumento unico per tutto il gruppo per ogni fase del ciclo di vita dei documenti, in modo da avere una documentazione uniforme e aderente agli standard e regole sotto riportate.

#### 3.1.3 Descrizione

Questa sezione fornisce i dettagli su come deve essere redatta, verificata, e approvata la documentazione. Tutte le norme descritte devono essere rispettate in pieno in tutti i documenti, sia interni che esterni, rilasciati durante il ciclo di vita del software.

#### 3.1.4 Attività

##### 3.1.4.1 Implementazione del processo

#### Ciclo di vita

Ogni documento viene creato basandosi sul seguente ciclo di vita:

- **stesura:** il documento viene creato da uno o più redattori seguendo le regole presenti in questo documento. Una volta terminata la stesura i redattori autorizzeranno l'avanzamento del documento all'attività successiva.
- **verifica:** viene effettuato da uno o più verificatori un controllo sul documento sia per quanto riguarda la stesura corretta, sia per un controllo semantico e sintattico seguendo le norme di progetto. Terminato il controllo se il documento dovesse essere corretto, i verificatori autorizzeranno il passaggio del documento alla fase successiva, in caso contrario tornerà alla prima fase, con le indicazioni delle parti da emendare.
- **approvazione:** è l'ultima fase del ciclo di vita del documento. In questa fase il responsabile autorizza o meno il rilascio il documento.

Il responsabile ha facoltà di intervenire, in ogni momento del ciclo di vita del documento, dando indicazioni se interpellato o per sua volontà.

### Documenti ufficiali

Si dicono ufficiali i documenti che:

- sono stati revisionati, verificati e approvati;
- sono gli unici rilasciabili all'esterno del gruppo di progetto.

I documenti ufficiali che saranno prodotti sono:

**Studio di fattibilità:** ha l'obiettivo di descrivere in modo breve ogni capitolato, mostrando quelli che per *DPCM 2077* possono essere punti di forza, aspetti interessanti o criticità. I destinatari di questo documento sono i membri del gruppo *DPCM 2077*.

**Piano di progetto:** stabilisce e documenta tutti gli aspetti del nostro progetto organizzandoli in modo da raggiungere i risultati nel modo più efficace ed efficiente possibile. Organizza le attività, ne analizza i rischi, e le associa ai membri del gruppo. I destinatari di questo documento sono il committente e il proponente oltre al gruppo *DPCM 2077*.

**Piano di qualifica:** espone e descrive i criteri di valutazione della qualità adottati dal gruppo. I destinatari di questo documento sono il committente e il proponente oltre al gruppo *DPCM 2077*.

**Analisi dei requisiti:** ha lo scopo di definire le funzionalità che il nostro progetto deve offrire e quindi i requisiti che devono essere soddisfatti. I destinatari di questo documento sono il committente e il proponente oltre al gruppo *DPCM 2077*.

**Verbali:** hanno lo scopo di riassumere in modo conciso i contenuti delle riunioni e soprattutto le decisioni prese da *DPCM 2077*.

**Glossario:** il documento è composto da un elenco ordinato di tutti i termini usati nella documentazione che il gruppo ritiene necessitano di una definizione esplicita, al fine di rimuovere ogni ambiguità. I destinatari di questo documento sono il committente e il proponente oltre al gruppo *DPCM 2077*.

#### 3.1.4.2 Sviluppo e design

##### Template LaTeX

Il gruppo ha deciso di utilizzare un template LaTeX per la stesura di tutti i documenti in modo da garantire uniformità tra questi.

##### Struttura generale dei documenti

Un file "main.tex" provvederà a raccogliere tutte le sezioni, pacchetti e comandi necessari per la sua compilazione. Tutti i documenti hanno una struttura predefinita e determinata.

## Frontespizio

Il frontespizio conterrà il logo e il nome del gruppo, il nome del documento, il nome del progetto al quale si riferisce e tutte le informazioni riguardanti il documento in sè quali:

- **versione:** versione attuale del documento;
- **uso:** destinazione d'uso del documento, che potrà essere "interno" o "esterno";
- **stato:** attuale stato del documento, che potrà essere "in redazione" o "approvato";
- **destinatari:** destinatari del documento;
- **redazione:** lista dei membri del gruppo che si sono occupati della stesura dello specifico documento;
- **verifica:** lista dei membri del gruppo che si sono occupati della verifica dello specifico documento;
- **approvazione:** nominativo del membro del gruppo che ha approvato il documento per il rilascio.

## Registro modifiche

Il registro delle modifiche consiste in una tabella contenente le informazioni riguardanti il ciclo di vita del documento. Più precisamente, la tabella riporta per ogni modifica:

- **versione:** versione del documento relativa alla modifica effettuata;
- **descrizione:** breve descrizione della modifica effettuata;
- **data:** data in cui la modifica è stata effettuata;
- **autore:** nominativo della persona che ha effettuato la modifica;
- **ruolo:** ruolo della persona che ha effettuato la modifica.

## Indice

L'indice delle sezioni fornisce al fruitore una visione complessiva della struttura del documento, permette di orientarsi tra i contenuti e di individuare la posizione delle varie parti. Ogni documento presenta un indice dei contenuti, subito dopo il registro delle modifiche.

## Verbali

I verbali sono suddivisi nelle seguenti sezioni:

- **introduzione:** essa contiene:
  - **luogo:** la piattaforma online in cui si svolge l'incontro;
  - **data:** data dell'incontro;

- **ora di inizio:** l'ora dell'inizio dell'incontro in formato HH:MM;
- **ora di fine:** l'ora della fine dell'incontro in formato HH:MM;
- **ordine del giorno:** consiste in una lista degli argomenti che il gruppo si è proposto di discutere durante l'incontro;
- **presenze:** contiene la lista dei presenti e degli assenti con eventuale giustificata;
- **svolgimento:** per ogni punto presente nell'ordine del giorno, viene riportato un riassunto di ciò che è stato trattato durante l'incontro;
- **tracciamento delle decisioni:** è un riepilogo in formato tabellare delle decisioni prese durante l'incontro; esso è composto da:
  - **codice:** del tipo "AAAA-MM-GG\_X.Y" dove la prima parte indica la data dell'incontro a cui fa riferimento seguita da un numero che indica il numero del verbale(X) ed un secondo numero che indica il punto all'ordine del giorno a cui si riferisce(Y);
  - **descrizione:** breve descrizione riassuntiva della decisione presa riguardante il punto dell'ordine del giorno.

## Glossario

Il *Glossario* segue la struttura principale di tutti gli altri documenti. Il contenuto è composto da una lista, che segue l'ordine lessicografico, di termini seguiti dalla loro spiegazione. I termini che si trovano nel glossario sono presenti all'interno di almeno uno dei documenti e ne viene data la spiegazione in quanto trattano argomenti ambigui e/o di poca comprensione, oppure hanno natura specifica o sono acronimi.

## Lettere

La *Lettera di presentazione* dovrà seguire il classico layout per lettere, il che implica la presenza dei mittenti e destinatari, il logo del gruppo e la lista di tutti i documenti rilasciati, nonché il preventivo per il progetto.

### 3.1.4.3 Norme tipografiche

#### Formati comuni

- **data:** viene utilizzato lo standard ISO 8601, esempio 2020-01-22, per la rappresentazione di tutte le date presenti della documentazione;
- **ora:** viene utilizzato il formato HH:MM;

## Grammatica e sintassi

- prediligere frasi brevi, con basso livello di subordinazione e intorno a un singolo argomento;
- attenzione alla concordanza fra tempi verbali: il tempo verbale deve essere coerente non solo nel singolo documento / capitolo che si scrive ma globalmente. Es. tendenzialmente non si utilizza

il presente in un verbale e dopo due righe il passato prossimo ma solo il passato prossimo (si registrano decisioni prese) né tantomeno si cambia il tempo verbale fra capitoli appartenenti allo stesso documento;

- l'uso della 'd' eufonica dovrebbe essere limitato ai casi di incontro della stessa vocale (es. ed ecco, ad andare, ad ascoltare, etc...). Per esempio non va bene ed organizzazione;
- attenzione agli accenti: perché e non perchè oppure perche';
- tutti i segni d'interpunzione seguono immediatamente la parola precedente e vanno separati con uno spazio da quella successiva;
- In genere le parentesi racchiudono un inciso nel discorso (cioè una parte che si può omettere, come questa). Inoltre, attenzione agli spazi: "esempio (x, y, z) esempio" e non esempio(x, y,z)esempio;
- acronimo tutto maiuscolo. Se si usa acronimo prima volta va specificato in esteso;
- nomi con l'iniziale maiuscola;
- Per gli elenchi puntati:
  - ogni voce di un elenco semplice (i cui elementi sono costituiti da un solo enunciato) comincia con l'iniziale minuscola e termina con il punto e virgola tranne l'ultima, seguita dal punto fermo;
  - ogni voce di un elenco complesso (in cui almeno uno degli elementi sia composto da più di un enunciato) comincia con l'iniziale maiuscola (anche dopo il segno di due punti) e termina con il punto fermo;
  - l'iniziale maiuscola è permessa se si fa riferimento a nomi di documentazione, nomi propri di persona, etc.;
  - non bisogna per forza uniformare tutti gli elenchi di un documento a criteri stabiliti a priori: l'importante è essere coerenti volta per volta.
- **grassetto**: per evidenziare parole importanti.

## Lettere maiuscole

Tutti i titoli vanno con iniziale maiuscola. Usare le maiuscole solo in caso di nomi propri e documenti (es Analisi dei requisiti. non analisi dei requisiti o Analisi dei Requisiti. È consentita l'uso della sigla AdR).

## Sigle

Il progetto prevede la redazione di un insieme di documenti, suddivisi in documenti interni e documenti esterni. Essi sono elencati di seguito con le rispettive sigle.

I documenti esterni sono:

- **Analisi dei requisiti - AdR**;
- **Piano di Progetto - PdP**;

- **Piano di Qualifica - PdQ;**
- **Manuale Utente - MU;**
- **Manuale Sviluppatore - MS.**

I documenti interni sono:

- **Glossario: G;**
- **Norme di Progetto: NdP;**
- **Studio di Fattibilità - StF.**

Per quanto riguarda il **Verbale - V** questo può essere sia interno sia esterno.

Le diverse fasi del progetto sono:

- **Revisione dei Requisiti - RR:** studio iniziale del C1;
- **Revisione di Progettazione - RP:** riguarda la definizione dell'architettura del software e della sua fattibilità;
- **Revisione di Qualifica - RQ:** riguarda la definizione dettagliata e la codifica del prodotto;
- **Revisione di Accettazione - RA:** se il prodotto soddisfa i requisiti del proponente, viene accettato e rilasciato.

### 3.1.5 Strumenti

#### 3.1.5.1 LaTeX

Lo strumento scelto per la scrittura di documenti è  $\text{\LaTeX}$ , un linguaggio che permette di scrivere documenti in modo ordinato, modulare e collaborativo.

#### 3.1.5.2 Google Drive

Si utilizza  $\text{GOOGLE DRIVE}_G$  per salvare file utili al gruppo di qualunque genere.

#### 3.1.5.3 Diagrammi

I software utilizzati per la realizzazione dei diagrammi sono i seguenti:

- **StarUML:** per la realizzazione di  $\text{DIAGRAMMI UML}_G$ ;
- **GanttProject:** applicazione open source per la realizzazione di  $\text{DIAGRAMMI DI GANTT}_G$  che mostrano lo svolgimento delle attività.

## 3.2 Gestione della Configurazione

### 3.2.1 Scopo

La gestione della configurazione definisce i principi normativi utili a predisporre il  $WORKSPACE_G$  per tutto il gruppo, semplificando e automatizzando la conservazione dei documenti e delle componenti software, che andranno a formare il prodotto finale.

### 3.2.2 Versionamento

#### 3.2.2.1 Codice di versione del documento

Ogni documento deve avere una storia, ricostruibile tramite le sue versioni. Ogni numero di versione è così composto:

**X.Y.Z**

- **X:** rappresenta una versione stabile del documento, resa tale dopo l'approvazione dal responsabile di progetto:
  - inizia da 0;
  - viene incrementato dal responsabile di progetto all'approvazione del documento.
- **Y:** rappresenta una versione parzialmente stabile del documento che è stata approvata parzialmente dal responsabile di progetto:
  - inizia da 0;
  - viene incrementato dal responsabile a ogni approvazione parziale. Per approvazione parziale si intende che una o più sezioni del documento sono pronte per il rilascio all'esterno del gruppo e che quindi non è necessario attendere l'approvazione totale del documento per poterle rilasciare.
  - quando viene incrementato X, viene riportato a 0.
- **Z:** rappresenta una modifica verificata del documento in attesa di approvazione:
  - inizia da 0;
  - viene incrementato dal redattore a ogni modifica;
  - quando viene incrementato X, viene riportato a 0.

#### 3.2.2.2 Versionamento con GitHub

Il repository fa uso di un Version Control System (VCS) di tipo distribuito sotto il motore *Git*, che permette la condivisione dal locale al remoto del proprio spazio di lavoro su un luogo comune. Attraverso l'utilizzo di un web browser, è possibile collegarsi a *GitHub* e controllare i file contenuti in un repository.



### 3.2.2.3 Repository

Ci sono due tipi di repository la cui struttura è identica e raccolgono i file sorgenti per la compilazione dei documenti, suddivisi tra esterni e interni:

- **locale:** ogni membro del gruppo lavora sui file clonati dal repository remoto nel proprio computer;
- **remoto:** pubblicato su *GitHub*, contiene il lavoro svolto da ogni componente e che viene condiviso con i membri di DPCM 2077.

### 3.2.2.4 Configurazione del workflow

Si è deciso di stabilire il seguente canone di  $WORKFLOW_G$ .

#### Utilizzo dei branch

- **master branch:** branch principale su cui vengono fatti i rilasci ad ogni  $MILESTONE_G$ ;
- **develop branch:** branch di sviluppo su cui viene fatta integrazione di nuove funzionalità concluse;
- **feature/... branch:** i puntini saranno sostituiti dal nome del documento a cui si riferisce il branch. Questo tipo di branch viene aperto al momento di effettuare una modifica ad una parte di codice, in modo tale da rendere le correzioni/implementazioni il più indipendenti possibile tra loro.

#### Utilizzo delle $PULL REQUEST_G$

Al termine dell'implementazione di una feature, sviluppata nel feature branch, viene aperta una Pull Request verso il develop branch: qui è possibile discutere di possibili correzioni su parte del codice introdotto, visualizzare tutte le modifiche effettuate e molto altro a supporto della gestione del codice. Una volta completate le modifiche richieste discusse, il responsabile può approvare la Pull Request introducendo le modifiche all'interno del ramo develop.

## 3.3 Garanzia della Qualità

### 3.3.1 Scopo

Lo scopo è di garantire che il prodotto e i servizi offerti rispettino gli obiettivi di qualità e che i bisogni del proponente siano soddisfatti.

### 3.3.2 Aspettative

Attraverso la gestione della qualità si intende ottenere:

- un prodotto software di qualità;
- una documentazione completa e facilmente comprensibile per tutti;
- dei processi che seguano dei punti ben specificati per l'analisi di qualità;
- una comunicazione chiara e semplice delle problematiche relative alla qualità tra i membri del team;
- una registrazione dei risultati ottenuti.

### 3.3.3 Descrizione

Per una presentazione dettagliata della gestione della qualità si fa riferimento al Piano di Qualifica v. 2.0.0.

### 3.3.4 Controllo qualità di prodotto

La qualità di prodotto viene garantita dai processi di verifica e validazione e dal rispetto delle norme.

- **Verifica:** accerta che l'esecuzione delle attività di processo siano corrette;
- **Validazione:** accerta che il prodotto rispecchi le aspettative, utilizzando un metodo sistematico, disciplinato e quantificabile;
- **Norme:** il prodotto deve rispettare le norme e gli strumenti descritti all'interno di questo documento.

Per ogni prodotto vengono descritti:

- le metriche da utilizzare;
- gli obiettivi da perseguire.

### 3.3.5 Controllo qualità di processo

La qualità di processo viene migliorata per tutto il ciclo di vita del software attraverso i principi di efficacia ed efficienza mirati al prodotto. Nello specifico definiamo quanto segue:

- **Efficacia:** si richiede un prodotto che soddisfi le richieste del proponente;
- **Efficienza:** i processi devono convergere con costi ridotti in termini di risorse a pari qualità di prodotto.

Per ogni processo vengono descritti:

- le metriche da utilizzare;
- gli intervalli entro i quali le misurazioni sono ritenute preferibili e accettabili.

### 3.3.6 Attività

Le tre attività principali sono:

- **Pianificazione:** porsi degli obiettivi, definire le strategie per raggiungerli e predisporre persone e risorse per raggiungere gli obiettivi nel modo più efficiente possibile;
- **Valutazione:** applicare i criteri misurando e monitorando i risultati;
- **Reazione:** sulla base dei risultati applicare le proprie strategie, criteri e piani.

### 3.3.7 Strumenti

Gli strumenti che garantiscono la qualità sono identificati negli standard **ISO-12207** e **ISO-9126**.

## 3.4 Verifica

### 3.4.1 Scopo

Il processo di verifica ha come scopo la realizzazione di un prodotto corretto, basandosi sulle richieste del committente/proponente e sulle norme definite in questo documento.

### 3.4.2 Aspettative

Lo svolgimento del processo di verifica sarà garantito seguendo determinati punti:

- definizione di criteri di accettazione;
- prescrizione delle attività di verifica con relativa documentazione;
- test di verifica;
- correzione di eventuali DIFETTI<sub>G</sub>.
- dopo la verifica il prodotto è in uno stato stabile;
- rende possibile la validazione.

### 3.4.3 Descrizione

Il processo di verifica prende in input ciò che è già stato prodotto e lo restituisce in uno stato conforme alle aspettative. Per ottenere tale risultato vengono effettuati processi di analisi e test.

### 3.4.4 Attività

#### 3.4.4.1 Analisi

Il processo di analisi si suddivide in:

- **Statica:** effettuata su documenti e codice, ha lo scopo di valutarne la conformità alle regole qui descritte, la mancanza di errori sintattici e grammaticali e la coesione dei componenti. Alcuni controlli sono da fare ad esempio sulle date, sulla forma dei verbi, sulla punteggiatura. Verranno usati due tipi di analisi statica:
  - **WALKTHROUGH<sub>G</sub>:** i vari componenti di DPCM 2077 analizzano gli oggetti nella loro totalità per cercare anomalie, senza sapere inizialmente se vi siano difetti;
  - **INSPECTION<sub>G</sub>:** i verificatori usano le liste di controllo per fare ispezione cercando errori specifici in parti specifiche.

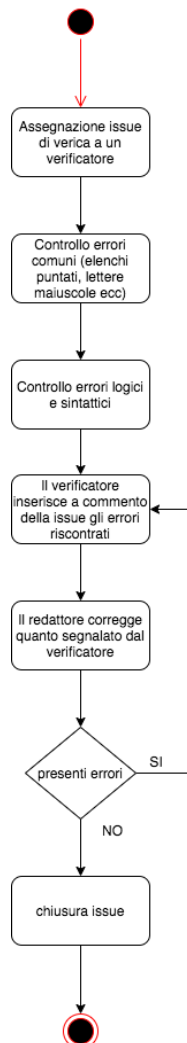


Figura 1: Verifica di un documento

- **Dinamica:** tecnica di analisi del prodotto software che richiede la sua esecuzione. Viene effettuata mediante dei test che verificano se il prodotto funziona o se ci sono anomalie.

### 3.4.4.2 Test

I test rappresentano l'attività fondamentale dell'analisi dinamica. Lo scopo dei test è dimostrare che il codice scritto funzioni correttamente e che non presenti errori. Per ogni test devono essere definiti i seguenti parametri:

- **ambiente:** sistema hardware e software sul quale verrà eseguito il test;
- **stato iniziale:** stato iniziale dal quale il test viene eseguito;
- **input:** dati in ingresso immessi;
- **output:** dati in uscita attesi;
- **istruzioni aggiuntive:** ulteriori istruzioni su come deve essere eseguito il test e su come interpretare i risultati ottenuti.

Dei test ben scritti, per definirsi tali, devono:

- essere ripetibili;
- specificare l'ambiente di esecuzione;
- identificare input e output richiesti;
- controllare la presenza di effetti indesiderati;
- fornire informazioni sui risultati dell'esecuzione sotto forma di file di log.

Per elencare le specifiche dei test si è scelta una rappresentazione tabellare contenente il codice del componente da testare, la descrizione dei test, il suo stato di avanzamento e il risultato del test. Di seguito la nomenclatura dello stato del test:

- **I:** il test è stato implementato;
- **NI:** il test non è stato implementato.

Di seguito la nomenclatura del risultato del test:

- **S:** il test si è concluso con successo;
- **F:** il test ha fallito.

Di seguito sono spiegati i diversi test del software.

### Test di unità

I test di unità si eseguono su unità di software. Questi test si concentrano sul funzionamento delle unità individuali. Dati gli input possibili in un'unità, si suddividono gli input in partizioni di equivalenza e si prova se gli input attesi danno gli output previsti. Questa tipologia di test verrà classificata nel seguente modo:

**TU[Identificativo]**

dove:

- **Identificativo:** numero progressivo univoco incrementale, inizia da 1.

### Test di integrazione

I test d'integrazione verificano la correttezza delle interfacce. Rappresentano l'estensione logica dei test d'unità e sono i predecessori dei test di sistema. Vengono svolti aggregando due o più unità già testate in un unico componente ed espandendo, di fatto, i test di moduli di un gruppo con quelli di altri gruppi. Ogni test prevede vi siano almeno un paio di componenti correlate per sottoporre a verifica tutte le unità che compongono un processo. Questa tipologia di test verrà classificata nel seguente modo:

**TI[Identificativo]**

dove:

- **Identificativo:** numero progressivo univoco incrementale, inizia da 1.

### Test di sistema

Una volta integrati tutti i suoi componenti il sistema viene testato nella sua interezza. Ci si concentra sulle interazioni tra le parti, sul comportamento delle caratteristiche del sistema e sulla copertura di tutte le funzionalità. In questa fase ci si assicura che il sistema rispetti tutte le specifiche definite nell'*Analisi dei Requisiti*. Questa tipologia di test verrà classificata nel seguente modo:

**TS[Priorità][Tipologia][Identificativo]**

dove:

- **Priorità:** indica la priorità del requisito associato al test:
  - **1:** obbligatorio, ossia irrinunciabile;
  - **2:** desiderabile, non strettamente necessario;
  - **3:** opzionale, ossia relativamente utile o contrattabile in corso d'opera.

- **Tipologia** : indica la tipologia del requisito associato al test:
  - *F*: funzionale;
  - *P*: prestazionale;
  - *Q*: qualitativo;
  - *V*: vincolo.
- **Identificativo**: numero progressivo univoco incrementale, inizia da 1.

### Test di regressione

Si effettua a seguito di una modifica del sistema e consiste nella riesecuzione dei test esistenti.

### Test di accettazione

Anche detto "test di collaudo", è simile al test di sistema per l'oggetto testato, ma viene eseguito con la partecipazione dei committenti. Verifica il prodotto e, in particolare, il soddisfacimento del cliente. Il superamento del test di collaudo garantisce che il software è pronto per essere rilasciato. Questa tipologia di test verrà classificata nel seguente modo:

**TA[Priorità][Tipologia][Identificativo]**

dove:

- **Priorità**: indica la priorità del requisito associato al test:
  - **1**: obbligatorio, ossia irrinunciabile;
  - **2**: desiderabile, non strettamente necessario;
  - **3**: opzionale, ossia relativamente utile o contrattabile in corso d'opera.
- **Tipologia** : indica la tipologia del requisito associato al test:
  - *F*: funzionale;
  - *P*: prestazionale;
  - *Q*: qualitativo;
  - *V*: vincolo.
  - **Identificativo**: numero progressivo univoco incrementale, inizia da 1.

### 3.4.5 Strumenti

#### 3.4.5.1 Verifica ortografica

Per la verifica ortografica dei documenti scritti in  $\text{\LaTeX}$  si utilizzerà lo strumento di correzione automatica presente negli editor Texmaker e TexStudio.

## 3.5 Validazione

### 3.5.1 Scopo

Lo scopo del processo di validazione consiste nel garantire che il prodotto soddisfi il compito per cui è stato creato e che quindi rispecchi le richieste del committente.

### 3.5.2 Aspettative

La corretta esecuzione di tale processo consente di individuare:

- una procedura di validazione;
- i criteri per la validazione;
- la conformità del prodotto finito.

### 3.5.3 Descrizione

Il processo di validazione consiste nell'identificare gli oggetti da validare e valutare che i risultati rispettino le aspettative.

### 3.5.4 Attività

Le attività di validazione sono:

1. eseguire i test sul prodotto;
2. i verificatori eseguono i test e stendono un resoconto sui risultati;
3. se i risultati non sono soddisfacenti si ritorna al primo punto e si rieseguono i test;
4. i risultati vengono inviati al proponente.



## 4 Processi Organizzativi

### 4.1 Gestione della comunicazione

In questa sezione vengono espone le norme che dovranno essere seguite per comunicare tra le varie parti (proponente, committenti, membri del gruppo DPCM 2077).

#### 4.1.1 Comunicazioni interne

Questa sezione riguarda le norme da rispettare per la comunicazione all'interno del gruppo DPCM 2077. Per le comunicazioni interne al gruppo è stato inizialmente adottata la nota applicazione TELEGRAM<sub>G</sub>. È stato creato un gruppo contenente i 7 membri di DPCM 2077. In seguito si è deciso di affiancare a Telegram anche l'applicazione DISCORD<sub>G</sub>. Discord consente di inviare messaggi, fare chiamate vocali e di condividere lo schermo. All'interno dell'applicazione sono stati creati vari sotto-canali riguardanti i vari documenti da redigere (e.g. glossario-aggiornamento, piano-di-progetto e norme-di-progetto) per poter scambiare messaggi solo con le persone coinvolte nella redazione di un certo documento. Per video-chiamate, oltre a ZOOM<sub>G</sub> è stato usato l'applicativo MEET<sub>G</sub> di Google per una maggiore semplicità di utilizzo.

Canali tematici predisposti all'interno di Discord:

- regole-general: per discutere di ciò che riguarda l'organizzazione generale del progetto, la scelta degli strumenti di lavoro, le decisioni da prendere in modo rapido, senza bisogno di indire una riunione o un incontro su Zoom / Meet;
- help-github: per poter chiedere aiuto nel caso di problemi durante il versionamento dei file;
- off-topic: dove discutere liberamente di tutto ciò che non concerne strettamente il progetto e/o lo sviluppo del prodotto;
- glossario-aggiornamento: per segnalare nuove parole da aggiungere al Glossario;
- piano-di-progetto: per poter discutere riguardo al documento esterno Piano di Progetto;
- norme-di-progetto: per poter discutere riguardo al documento interno Norme di Progetto;
- analisi-dei-requisiti: per poter discutere riguardo al documento esterno Analisi dei requisiti.

I membri del gruppo DPCM 2077 sono tenuti a comunicare tra loro, rispettando le tematiche dei canali preposti. Nel caso volessero portare all'attenzione di un particolare membro del gruppo un certo messaggio sono tenuti ad usare il comando @userName. Invece usando il comando @everyone è possibile portare il messaggio all'attenzione di tutti. Le riunioni possono essere svolte solo tramite canali telematici, a causa del propagarsi della pandemia, e quindi non in presenza fisica.

#### 4.1.2 Comunicazioni esterne

Questa sottosezione riguarda le norme alle quali attenersi per comunicare con soggetti esterni a DPCM 2077. Soggetti esterni individuati:

- la proponente IMOLA INFORMATICA<sub>G</sub>. È possibile comunicare con il Dr. Lorenzo Patera e il Dr. Luca Cappelletti che hanno contribuito a creare il capitolato d'appalto;
- i committenti Prof. Tullio Prof. Vardanega, ai quali verrà fornita tutta la documentazione necessaria a sostenere con successo le varie revisioni di progetto, e con i quali si intende creare un rapporto che migliori di continuo i processi e le strategie adottate.

#### 4.1.2.1 Comunicazioni esterne scritte

Le comunicazioni esterne scritte devono avvenire attraverso l'utilizzo dell'indirizzo email del gruppo: **dpcm2077@gmail.com**. Nel caso di un messaggio da inviare rivolto a più destinatari, nel campo "A:" va indicato il destinatario principale, e nel campo "CC:" vanno indicati tutti gli altri destinatari (mettere in copia). Le email dovranno essere scritte dal Responsabile di Progetto che dovrà avere cura di includere il logo del gruppo e di apporre la propria firma.

## 4.2 Riunioni

Questa sezione ha lo scopo di definire le stesse e le loro modalità di svolgimento. Nelle riunioni è il responsabile che convoca la riunione con scadenza settimanale, stende l'o.d.g, assegnando la giusta priorità ai vari punti da discutere e conduce / modera la discussione. Se qualcuno del gruppo vuole proporre un argomento contatta i responsabili prima della riunione, che lo inseriranno nell'o.d.g e gli daranno la priorità corretta e lo spazio se c'è. Un componente nominato segretario redige solo il verbale. In Google drive è stato creato un o.d.g. temporaneo dove i componenti di DPCM 2077 possono vedere l'o.d.g. della riunione in anticipo. Inoltre è stato creato un documento dove raccogliere le decisioni prese nel corso della settimana, tra una riunione e la successiva, per non perderne traccia. Esse saranno validate alla prima riunione utile. È compito del responsabile tenere aggiornati l'o.d.g. temporaneo e il tracciamento delle decisioni settimanali.

#### 4.2.1 Verbali di riunione

Nello svolgimento delle riunioni è compito del segretario nominato (a rotazione) redigere il verbale di riunione secondo le seguenti indicazioni:

- **introduzione**, questa sezione contiene:
  - luogo: ad esempio, videoconferenza su Meet;
  - data: nel formato AAAA-MM-GG, ad esempio 2020-12-14;
  - ora di inizio: nel formato ventiquattro ore, ad es. 15:30;
  - ora di fine: nel formato ventiquattro ore, ad es. 15:30;
  - ordine del giorno: punti discussi sotto forma di elenco puntato.
- **Presenze**
  - presenti: nominativi dei componenti presenti;
  - assenti: nominativi dei componenti eventualmente assenti;

- **svolgimento:** contiene le annotazioni circa gli argomenti discussi e non discussi tra quelli presenti nell'Ordine del giorno, le decisioni prese o non prese e le motivazioni che hanno portato a tali decisioni.
- **tracciamento delle decisioni:** riepilogo in forma tabellare delle decisioni prese nel corso della riunione;

**Nomenclatura dei verbali:** i verbali interni dovranno avere come nome: **VI\_AAAA-MM-GG\_v1.0.0**, ad esempio VI\_2021-01-11\_v1.0.0 e quelli esterni: **VE\_AAAA-MM-GG\_v1.0.0**. v1.0.0 è un esempio di versione.

**Modalità di conservazione dei verbali:** i verbali sono redatti in  $\text{\LaTeX}$ , i verbali interni sono conservati nella cartella **interni** nel repository `GITHUBG` contenente la documentazione, e quelli esterni nella cartella **esterni**.

#### 4.2.2 Riunioni interne

Alle riunioni interne partecipano solo i componenti di DPCM 2077. In un'epoca non pandemica gli incontri potevano avvenire anche di persona in un luogo, invece adesso si utilizzano gli strumenti Zoom / Meet per poter effettuare videochiamate.

##### **Compiti del Responsabile di progetto durante le riunioni:**

- fissare la data delle riunioni previa consultazione degli altri componenti del gruppo Discord;
- ascoltare i consigli dei vari componenti del gruppo (su tecnologie oppure documenti) e decidere se tenerli in considerazione, quindi approvarli, oppure respingerli;
- cercare di rispettare l'ordine del giorno, cioè i punti da discutere;
- approvare il verbale redatto dal componente incaricato (segretario);
- nel caso di annullamento di una riunione, comunicarlo tempestivamente utilizzando Telegram / Discord.



Figura 2: Organizzazione riunione interna

#### Compiti dei partecipanti durante le riunioni:

- cercare di partecipare sempre attivamente alle riunioni: esporre le proprie perplessità affinché i problemi vengano risolti subito;
- cercare di essere presenti all'ora stabilita, nel caso di ritardi o impossibilità di presentarsi avvisare il Responsabile di Progetto;
- comportamento corretto e linguaggio adeguato al contesto;
- cercare di evitare rumori di sottofondo durante le riunioni telematiche per non creare disturbo agli altri componenti.

#### 4.2.3 Riunioni esterne

Una riunione è esterna quando a essa, oltre ai componenti del gruppo, partecipano il committente e/o il proponente. Le riunioni esterne avvengono sempre in modalità telematica usando Zoom / Meet. Nel caso di un ritorno alle modalità in presenza potrebbero essere fatte in locali universitari previo consenso da parte dei docenti, non nella sede dell'azienda essendo molto distante da Padova. Per poter avere una riunione con il committente / proponente avvisarlo in anticipo di almeno 5 giorni. Per poter avvisare il committente utilizzare la mail, per avvisare il proponente utilizzare Telegram.



Figura 3: Organizzazione riunione esterna

### 4.3 Ruoli di progetto

Tutti i ruoli saranno ricoperti da ciascun componente del gruppo in rotazione, secondo un calendario concordato, in modo che ogni membro possa assumere almeno una volta ciascuno di essi. All'incirca un ruolo dovrà essere mantenuto per circa 14 giorni, tempo necessario per poterlo ricoprire al meglio (per esempio il programmatore deve prima studiare il linguaggio e solo poi diventa effettivamente produttivo, quindi troppo pochi giorni non bastano). Nel documento Piano di Progetto, vengono organizzate e pianificate le attività assegnate ai specifici ruoli previsti nell'attività di progetto. I Ruoli di Progetto sono:

- **Responsabile di progetto;**
- **Amministratore;**
- **Analista;**
- **Progettista;**
- **Programmatore;**
- **Verificatore;**

#### 4.3.1 Responsabile di progetto

Il Responsabile di Progetto, o “Project Manager”, è una figura essenziale e partecipa al progetto per tutta la sua durata.

- Governa il team e rappresenta il progetto verso l'esterno
  - accentra la responsabilità di scelta e approvazione;
- Ha responsabilità su:
  - pianificazione;
  - gestione delle risorse umane;
  - controllo, coordinamento e relazioni esterne;
  - approvazione della documentazione;
  - approvazione dell'offerta economica;
  - determinare e verificare le condizioni di completezza, ovvero quando il prodotto contiene tutti i requisiti richiesti dal committente e individuati dal gruppo.
- Deve avere conoscenze e capacità tecniche
  - per valutare rischi, scelte, alternative.

#### 4.3.2 Amministratore di progetto

L' Amministratore è quella figura che ha il controllo sull'ambiente di lavoro. Inoltre ha diretta responsabilità sull'efficienza e sulla capacità operativa dell'ambiente di lavoro. Ha in capo la redazione e la manutenzione migliorativa delle Norme di Progetto. Egli:

- controlla le versioni e le configurazioni del prodotto;
- risolve i problemi legati alla gestione dei processi;
- fornisce procedure e strumenti di monitoraggio/segnalazione, in modo da garantire un corretto controllo di qualità;
- individua strumenti atti all'automazione dei processi, cioè studia e ricerca strumenti che riducano il più possibile l'impiego di risorse umane;
- gestisce il versionamento della documentazione del progetto e della sua archiviazione.

#### 4.3.3 Analista

L' Analista si occupa dell'analisi dei problemi e del DOMINIO APPLICATIVO<sub>G</sub>. Svolge un ruolo fondamentale ai fini della riuscita del progetto, nonostante vi partecipi per un periodo di tempo limitato. Non segue il progetto fino alla consegna. L'Analista redige l'Analisi dei requisiti e lo Studio di Fattibilità. Egli:

- si occupa dei documenti di Analisi dei requisiti e di Studio di Fattibilità;
- conosce il dominio del problema e ha esperienza professionale;
- ha molta influenza sul successo del progetto;
- analizza il dominio applicativo: gli utenti, l'ambiente d'uso.

#### 4.3.4 Progettista

Il Progettista è la figura che si occupa delle scelte architettureali del progetto e ne influenza gli aspetti tecnici e tecnologici, avendo competenze tecniche e tecnologiche aggiornate. Utilizzando le attività svolte dall'Analista, il Progettista ha il compito di trovare una soluzione attuabile, comprensibile, chiara e motivata. Egli:

- produce una soluzione attuabile e chiara;
- sviluppa l'architettura secondo un insieme di best practice per garantire coerenza e consistenza;
- rende facilmente mantenibile il progetto;
- definisce una struttura che abbia un basso grado di accoppiamento; un forte accoppiamento non è mai buona norma a meno che non si tratti di coesione;
- sviluppa un'architettura solida, sicura e flessibile, in modo che rimanga funzionante in caso di malfunzionamenti e sia modificabile a basso costo;
- applica soluzioni note e ottimizzate.

#### 4.3.5 Programmatore

Il Programmatore è il responsabile della codifica del progetto e delle componenti di supporto, che serviranno per effettuare le prove di verifica e validazione sul prodotto. Partendo dall'attività del Progettista, il Programmatore realizza l'architettura già definita con l'unica responsabilità di svolgere al meglio il lavoro già progettato. Egli:

- implementa in maniera precisa e scrupolosa le soluzioni generate dal Progettista;
- scrive codice sorgente altamente e facilmente manutenibile;
- una volta terminata la codifica, se il prodotto realizzato non è di facile utilizzo può redigere un Manuale Utente;
- si occupa anche del versionamento e della documentazione del codice sorgente.

#### 4.3.6 Verificatore

È una figura presente per tutta la durata del progetto. Ha competenze tecniche, esperienza professionale e conoscenza del WAY OF WORKING. Redige la parte retrospettiva del Piano di Qualifica che illustra l'esito e la completezza delle verifiche e delle prove effettuate secondo il piano. Egli:

- assicura che nel corso del progetto vengano rispettate le norme redatte in questo documento di Norme di Progetto;
- assicura che ogni stadio del ciclo di vita del prodotto sia conforme al Piano di Qualifica.

## 4.4 Formazione del personale

Per quanto riguarda la formazione, i membri del gruppo DPCM 2077 devono procedere in modo autonomo con lo studio individuale delle tecnologie che verranno utilizzate nel corso del progetto, prendendo come materiale di riferimento quello indicato nella sezione **Riferimenti Informativi**. Si è comunque deciso di avere all'interno del Team un esperto di Git e  $\text{\LaTeX}$ , in modo che nel caso di ambiguità o difficoltà nell'utilizzo i componenti possono direttamente rivolgersi a queste persone. Con lo studio autonomo poi si possono aiutare i componenti che si trovano magari in difficoltà con qualche strumento o tecnica, dando origine così a un miglioramento continuo per tutti e ad un'efficienza e qualità maggiore nelle attività.

## 4.5 Ambiente di lavoro

### 4.5.1 Ticketing

Per assegnare le varie attività da svolgere tra i componenti del gruppo DPCM 2077 si è scelto di ricorrere all'ISSUE TRACKING SYSTEM<sub>G</sub> (ITS) di GitHub. Quando una  $\text{ISSUE}_G$  viene assegnata a un componente del gruppo, automaticamente gli viene inviata una e-mail. Quando si va a creare una issue, bisogna compilare i seguenti campi:

- **title**: indica il titolo dell'attività da svolgere o criticità da risolvere, nel caso della stesura di un documento il titolo deve corrispondere al nome del documento;
- **leave a comment**: area di testo dove è possibile specificare meglio la descrizione della issue;
- **assignees**: indica il componente o i componenti ai quali viene assegnato il task da svolgere o la criticità da risolvere;
- **labels**: indica le etichette che vengono associate alla issue, per esempio: bug, documentation, help wanted;
- **milestone**: specifica a che Milestone è associata la issue che si sta creando.





Figura 4: Assegnazione di un ticket

#### 4.5.2 Coordinamento attività

**Versionamento:** per il versionamento si è deciso di utilizzare Git, poiché conosciuto da tutti i membri del gruppo DPCM 2077 grazie anche all'insegnamento offerto nel Corso di Laurea **Tecnologie Open Source** frequentato da tutti i componenti. Con Git è incentivato lo sviluppo su  $BRANCH_G$  diversi che possono essere locali o condivisi. La maggior parte delle operazioni viene fatta in locale, i requisiti primari sono la velocità e le performance. Possono essere adottati diversi tipi di  $WORKFLOW_G$  e ogni commit è identificato da un ID che ne garantisce l'integrità. Non è possibile cambiare un commit senza modificare l'ID del commit stesso e dei i commit successivi. Si è stabilito inoltre di appoggiarsi al servizio di hosting GitHub per la sua interfaccia intuitiva e per l' Issue Tracking System integrato che offre molteplici funzionalità

**Pianificazione delle attività:** per la pianificazione delle attività il gruppo ha deciso di utilizzare il software **GANTTPROJECT<sub>G</sub>**. È un programma completamente scritto in linguaggio Java e sotto licenza GPL, quindi gratuito e liberamente utilizzabile, che permette di pianificare ogni tipo di

progetto o attività utilizzando la tecnica di Gantt. Può essere utilizzato sui principali sistemi operativi, quali: Windows, macOS e Linux. È quindi possibile:

- **creare l'elenco delle attività per il progetto:** data, ora di inizio e priorità;
- **determinare i tempi necessari per le singole attività;**
- **salvare spesso il progetto;**
- **annullare le ultime operazioni in caso di errore;**
- **generare diagrammi PERT:** per le dipendenze temporali fra attività e evidenziando le loro criticità;

**Continuous Integration:** per la Continuous Integration il team ha preso in considerazione i seguenti strumenti: **TRAVISCI<sub>G</sub>**, **JENKINS<sub>G</sub>** e **GITHUB ACTIONS<sub>G</sub>**; Si è deciso in seguito di utilizzare GitHub Actions in quanto direttamente integrato nel sistema di versionamento utilizzato e perchè GitHub offre modelli di workflow CI per una varietà di linguaggi e framework. Non è stato immediatamente scartato TravisCI in quanto già conosciuto da parte dei membri del team, ma non integrato su GitHub tanto quanto GitHub Actions. La Continuous Integration che utilizza GitHub Actions offre workflow che possono creare il codice nel repository ed eseguire i test. I workflow possono essere eseguiti su macchine virtuali ospitate da GitHub o su macchine ospitate personalmente. GitHub esegue i test di CI e fornisce i risultati di ogni test nella richiesta **PULL<sub>G</sub>**, così è possibile vedere se la modifica di un ramo introduce un errore. Quando tutti i test CI in un workflow vengono superati, le modifiche inviate sono pronte per essere riviste da un membro del team. Quando un test fallisce, una delle modifiche apportate potrebbe aver causato il fallimento.

### 4.5.3 Documentazione

**LaTeX:** è un programma professionale di composizione tipografica multilingua, multiplatforma e completamente gratuito. Permette di produrre documenti in formato pdf di elevata qualità, adatti sia per la stampa digitale, sia per la pubblicazione e la consultazione on-line. Altri strumenti come Google Docs e LibreOffice per la scrittura di documenti sono stati scartati in quanto non adeguati per scrivere documenti ufficiali per il progetto.

**Editor:** l'editor consigliato per scrivere la documentazione con **LaTeX** è **TexStudio** in quanto software libero aggiornato e multiplatforma. Per piattaforma macOS è invece disponibile anche l'editor **TeXShop**. Entrambi i programmi una volta compilati i file **.tex** producono in automatico un file **.pdf** con il documento.

**Diagrammi UML:** per modellare con UML è stato deciso di utilizzare il software **StarUML**. È uno strumento per creare diagrammi delle classi e altri tipi di diagrammi secondo Unified Modeling Language (UML). Viene anche utilizzato per la generazione automatica di codice Java a partire dalla sua rappresentazione UML; è possibile, inoltre, effettuare l'operazione inversa (reverse engineer) del codice Java sorgente per produrre il diagramma UML corrispondente.

### 4.5.4 Ambiente di sviluppo

**Sistemi Operativi:** i membri del gruppo DPCM 2077 potranno lavorare sui principali sistemi operativi presenti nel mercato, cioè: Windows, macOS e Linux. I principali strumenti e programmi richiesti sono ormai presenti in tutti e tre i sistemi operativi.

### Ambienti integrati di sviluppo

1. **WebStorm:** WebStorm è un IDE di grande valore per chi sviluppa con Javascript per alcuni semplici motivi, elenchiamo i principali:

- WebStorm è pieno di funzioni interessanti di auto-completamento;
- controllo della sintassi e adeguamento alle varie versioni del linguaggio;
- WebStorm è in grado di supportare TypeScript.

Ci sono anche altri lati positivi di WebStorm, ad esempio quelli che sono inerenti alle funzioni di refactoring del codice e ovviamente all'integrazione con i sistemi di versioning.

2. **PyCharm:** È particolarmente utile per chi sviluppa in Python perché integra diverse funzionalità utili nel development, alcuni esempi:

- il completamento della sintassi delle istruzioni;
- l'help online;
- l'evidenziazione degli errori di sintassi durante lo sviluppo;
- la console di Python in una finestra dell'editor.

## 4.6 Strumenti organizzativi

Il gruppo nel corso del progetto ha utilizzato e utilizzerà i seguenti strumenti:

- **Telegram:** strumento di messaggistica utilizzato inizialmente per la gestione del gruppo;
- **GitHub:** per il versionamento e il salvataggio in remoto di tutti i file riguardanti il progetto;
- **Google Drive:** utilizzato per il salvataggio di grafici, tabelle e per una gestione generale del progetto;
- **Discord:** utilizzato come canale principale per le comunicazioni tra i membri del gruppo.

## A Standard di Qualità

### A.1 ISO/IEC 9126

ISO/IEC 9126 è uno standard utilizzato a livello internazionale per valutare la qualità del software. È suddiviso in quattro parti che sono:

- **Modello della qualità del software;**
- **Metriche per la qualità esterna;**
- **Metriche per la qualità interna;**
- **Metriche per la qualità in uso.**

#### A.1.1 Metriche per la qualità interna

La qualità interna, più precisamente le metriche interne, è specificata nella norma *ISO/IEC 9126-3* e si applica al software non eseguibile durante le fasi di progettazione e codifica. Le misure effettuate permettono di prevedere il livello di qualità esterna e in uso del prodotto finale pronto per il rilascio, poiché gli attributi interni influiscono su quelli esterni e quelli in uso. Le metriche interne permettono di catturare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il software eseguibile.

#### A.1.2 Metriche per la qualità esterna

Le metriche esterne, specificate nella norma *ISO/IEC 9126-2*, misurano i comportamenti del software sulla base dei test, dall'operatività e dall'osservazione del prodotto durante la sua esecuzione, in funzione degli obiettivi stabiliti in un contesto tecnico rilevante o di business. Sono metriche applicabili al software in esecuzione che ne misurano il comportamento attraverso vari test, in funzione degli obiettivi precedentemente stabiliti.

#### A.1.3 Metriche per la qualità in uso

La qualità in uso rappresenta il punto di vista dell'utente sul software. Sono metriche applicabili solo al prodotto finito e in uso in condizioni reali, cioè quando il prodotto sarà usato effettivamente dal cliente. Il livello di qualità in uso è raggiunto quando sia il livello di qualità esterna sia il livello di qualità interna sono stati raggiunti. Le norme *ISO/IEC 9126-4* forniscono esempi di metriche da utilizzare per la misurazione della qualità rispetto ai tre punti di vista (interno, esterno, in uso). La qualità in uso, quindi, permette di abilitare specifici utenti a ottenere determinati obiettivi con efficacia, produttività, sicurezza e soddisfazione.

- **Efficacia:** la capacità del software di mettere in grado gli utenti di raggiungere gli obiettivi specificati con accuratezza e completezza;

- **Produttività:** la capacità di mettere in grado gli utenti di spendere una quantità di risorse appropriate in relazione all'efficacia ottenuta in uno specificato contesto d'uso;
- **Soddisfazione:** è la capacità del prodotto software di soddisfare gli utenti;
- **Sicurezza:** rappresenta la capacità del prodotto software di raggiungere accettabili livelli di rischio di danni a persone, al software, ad apparecchiature o all'ambiente operativo d'uso.

#### A.1.4 Modello della qualità del software

Il modello di qualità stabilito nella prima parte dello standard, *ISO/IEC 9126-1*, è classificato da sei caratteristiche generali (funzionalità, affidabilità, efficienza, usabilità, manutenibilità, portabilità) e varie sottocaratteristiche misurabili attraverso delle metriche, fornite in tre relazioni tecniche, quali *ISO/IEC 9126-2*, 3 e 4.

##### A.1.4.1 Funzionalità

La Funzionalità è la capacità di un prodotto software di fornire funzioni che soddisfano i requisiti stabiliti nell'ANALISI DEI REQUISITI<sub>D</sub>, necessarie per operare sotto condizioni specifiche. Le caratteristiche sono:

- **appropriatezza:** rappresenta la capacità del prodotto software di fornire un adeguato insieme di funzioni per i compiti e obiettivi prefissati dall'utente;
- **accuratezza:** la capacità del prodotto software di fornire i risultati concordati o i precisi effetti richiesti;
- **interoperabilità:** è la capacità del prodotto software di interagire e operare con uno o più sistemi specificati;
- **conformità:** la capacità del prodotto software di aderire a standard, convenzioni e regolamentazioni;
- **sicurezza:** la capacità del prodotto software di proteggere informazioni e dati negando in ogni modo che persone o sistemi non autorizzati possano accedervi o modificarli.

##### A.1.4.2 Affidabilità

L'Affidabilità è la capacità del prodotto software di mantenere uno specifico livello di prestazioni quando usato in date condizioni e per un dato periodo di tempo. Le caratteristiche sono:

- **maturità:** capacità di un prodotto software di evitare che si verifichino errori, malfunzionamenti o anomalie nell'utilizzo del prodotto;
- **tolleranza agli errori:** è la capacità di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o uso non corretto del prodotto software;
- **recuperabilità:** è la capacità di un prodotto di ripristinare il livello appropriato di prestazioni e di recupero delle informazioni rilevanti, in seguito a un malfunzionamento.

- **aderenza:** è la capacità di aderire a standard, regole e direttive inerenti all'affidabilità.

#### A.1.4.3 Efficienza

L'efficienza è la capacità di fornire appropriate prestazioni relativamente alla quantità di risorse usate. Le caratteristiche sono:

- **comportamento rispetto al tempo:** capacità di fornire adeguati tempi di risposta, elaborazione e velocità di attraversamento, sotto determinate condizioni;
- **utilizzo delle risorse:** capacità di raggiungere gli obiettivi prefissati consumando le risorse minime indispensabili;
- **conformità:** capacità di aderire a standard, specifiche e regole sull'efficienza.

#### A.1.4.4 Usabilità

L'usabilità è la capacità del prodotto software di essere compreso, usato e benaccetto dall'utente, quando usato in tutte le condizioni possibili. Le caratteristiche sono:

- **comprensibilità:** capacità di esprimere la facilità di comprensione dei concetti del prodotto, mettendo in grado l'utente di comprendere se il software è adatto per i propri scopi;
- **apprendibilità:** capacità di ridurre l'impegno richiesto agli utenti per imparare a usare la sua applicazione;
- **operabilità:** capacità di mettere in condizione gli utenti di usare il prodotto per i propri scopi e controllarne l'uso;
- **attrattiva:** capacità del software di essere piacevole per l'utente che ne fa uso;
- **conformità:** capacità del software di aderire a standard o direttive relativi all'usabilità.

#### A.1.4.5 Manutenibilità

La manutenibilità è la capacità del software di essere modificato, subendo correzioni, miglioramenti o adattamenti. Le caratteristiche sono:

- **analizzabilità:** rappresenta la facilità con la quale è possibile analizzare il codice per localizzare un errore nello stesso;
- **modificabilità:** capacità del prodotto software di permettere l'implementazione di una specificata modifica;
- **stabilità:** capacità del software di evitare condizioni inattese derivanti da modifiche errate;
- **testabilità:** capacità di essere facilmente testato per validare le modifiche apportate al software.

#### A.1.4.6 Portabilità

La portabilità è la capacità del software di essere trasportato da un ambiente di lavoro a un altro, hardware o software (come, per esempio, il sistema operativo).

- **adattabilità:** capacità del software di essere adattato per differenti ambienti operativi senza dover modificare il software considerato;
- **installabilità:** capacità del software di essere installato in uno specificato ambiente;
- **conformità:** capacità del prodotto software di aderire a standard e direttive relative alla portabilità;
- **sostituibilità:** capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti nello stesso ambiente.