

# SWEG

## Software Engineering Group

*swegsoftware@gmail.com*

### Norme di Progetto

#### Informazioni sul documento

Redattori:	Andrea M.	Davide S.	Gabriel R.
Verificatori:	Milo S.	Mircea P.	Marco B.
Approvazione:	Davide M.		
Destinatari:	T. Vardanega	R. Cardin	Zero12
Uso:	Interno		
Versione:	2.0.0		

## Registro dei Cambiamenti - Changelog

<i>Versione</i>	<i>Data</i>	<i>Autore</i>	<i>Verificatore</i>	<i>Dettaglio</i>
2.0.0	2023-05-17	Davide Milan	Gabriel Rovesti	Approvazione
1.8.0	2023-04-26	Gabriel Rovesti	Andrea Meneghello	Sistemazione indici e correzione varie sezioni e contenuti di massima
1.6.0	2023-04-25	Andrea Meneghello	Gabriel Rovesti	Completamento sezioni standard e varie fix minori
1.5.0	2023-04-10	Andrea Meneghello	Gabriel Rovesti	Scrittura sezione standard B
1.4.0	2023-03-31	Andrea Meneghello	Gabriel Rovesti	Scrittura sezione standard A
1.3.0	2023-02-15	Andrea Meneghello	Gabriel Rovesti	Definizioni e aggiornamento
1.2.0	2023-01-20	Andrea Meneghello	Mircea Plamadeala	Definizione sezioni nuove e creazione indici nuovi contenuti
1.1.0	2023-01-10	Andrea Meneghello	Mircea Plamadeala	Correzione e aggiornamenti da feedback
1.0.0	2022-12-21	Gabriel Rovesti	Andrea Meneghello	Approvazione
0.9.0	2022-12-18	Gabriel Rovesti	Andrea Meneghello	Scrittura sezioni finali documento (4.1.7, 4.3, 4.4) e correzione sezioni
0.8.0	2022-12-16	Mircea Plamadeala	Davide Sgrazzutti	Fix minori e correzione sezione 4
0.7.5	2022-12-14	Andrea Meneghello	Marco Bernardi	Fix minori e correzione sezione 3
0.7.0	2022-12-13	Andrea Meneghello	Davide Sgrazzutti	Scrittura sezione 4.1.5 e 4.2 e definizione struttura finale sez. 4
0.6.5	2022-12-12	Davide Sgrazzutti	Mircea Plamadeala	Sviluppo sezione 4, correzione e fix minori
0.6.0	2022-12-06	Davide Sgrazzutti	Marco Bernardi	Correzione termini e aggiunta riferimenti a glossario
0.5.0	2022-12-01	Andrea Meneghello	Davide Sgrazzutti	Definizione sezioni 3.3, 3.4 e 3.5

0.4.0	2022-11-30	Andrea Meneghello	Milo Spadotto	Scrittura e completamento sezione 3 e 3.1
0.3.0	2022-11-30	Andrea Meneghello	Milo Spadotto	Cambio struttura sez 2, completamento sezione 2.1/.2.2
0.2.0	2022-11-27	Gabriel Rovesti	Milo Spadotto	Inizio crittura sezione 2 e sottosezioni 2.1/2.2
0.1.0	2022-11-24	Andrea Meneghello	Milo Spadotto	Scrittura sezione 1, stesura sottosezioni e inizio sezione 3

# Sommario

<b>Sommario .....</b>	<b>3</b>
<b>1 Introduzione .....</b>	<b>7</b>
1.1 Scopo del documento .....	7
1.2 Scopo del prodotto .....	7
1.3 Glossario .....	8
1.4 Maturità e miglioramenti .....	8
1.5 Riferimenti .....	8
1.5.1 Riferimenti normativi .....	8
1.5.2 Riferimenti informativi .....	8
<b>2 Processi Primari .....</b>	<b>10</b>
2.1 Fornitura .....	10
2.1.1 Descrizione, scopo e aspettative .....	10
2.1.2 Valutazione dei capitolati .....	10
2.1.3 Piano di Progetto .....	10
2.1.4 Piano di Qualifica .....	11
2.1.5 Collaudo e rilascio .....	11
2.2 Sviluppo .....	11
2.2.1 Descrizione, scopo e aspettative .....	11
2.2.2 Analisi dei requisiti .....	12
2.2.2.1 Denominazione e legenda .....	12
2.2.2.1.1 Struttura Casi d'uso .....	12
2.2.2.1.2 Denominazione Casi d'uso .....	12
2.2.2.1.3 Struttura Requisiti .....	12
2.2.2.1.4 Denominazione Requisiti .....	13
2.2.3 Progettazione .....	13
2.2.3.1 Descrizione, scopo e aspettative .....	13
2.2.4 Codifica .....	14
2.2.4.1 Descrizione, scopo e aspettative .....	14
<b>3 Processi di Supporto .....</b>	<b>14</b>
3.1 Documentazione .....	14
3.1.1 Descrizione, scopo e aspettative .....	14
3.1.2 Ciclo di vita .....	15
3.1.3 Template .....	15
3.1.4 Documenti prodotti .....	15
3.1.4.1 Formali .....	15
3.1.4.2 Informali .....	16
3.1.5 Struttura del documento .....	16
3.1.5.1 Prima pagina .....	16
3.1.5.2 Registro dei Cambiamenti - Changelog .....	16
3.1.5.3 Indice .....	17

3.1.5.4 Piè di pagina .....	17
3.1.5.5 Verbali .....	17
3.1.6 Norme tipografiche.....	17
3.1.6.1 Nome del file .....	17
3.1.6.2 Stile del testo.....	17
3.1.6.3 Glossario.....	18
3.1.6.4 Altre Norme Tipografiche.....	18
3.1.7 Elementi grafici .....	18
3.1.7.1 Tabelle .....	18
3.1.7.2 Immagini.....	19
3.1.7.3 Grafici .....	19
3.1.8 Strumenti .....	20
3.1.8.1 Microsoft Word .....	20
3.1.8.2 Google Docs.....	20
3.1.8.3 UML .....	20
3.2 Gestione della Configurazione .....	20
3.2.1 Descrizione, scopo e aspettative .....	20
3.2.2 Versionamento.....	20
3.2.3 Repository .....	21
3.2.3.1 Tecnologie .....	21
3.2.3.2 Struttura repository.....	21
3.3 Gestione della qualità.....	22
3.3.1 Descrizione, scopo e aspettative .....	22
3.3.2 Piano di Qualifica .....	22
3.3.3 Denominazione metriche.....	22
3.4 Verifica.....	23
3.4.1 Descrizione, scopo e aspettative .....	23
3.4.2 Tipi di Verifica .....	23
3.4.2.1 Analisi statica.....	23
3.4.2.2 Analisi Dinamica .....	24
3.4.2.3 Denominazione dei test .....	24
3.4.2.4 Verifica della Documentazione .....	25
3.5 Validazione .....	25
3.5.1 Descrizione, scopo e aspettative .....	25
<b>4 Processi Organizzativi .....</b>	<b>25</b>
4.1 Gestione dei processi .....	25
4.1.1 Scopo.....	25
4.1.2 Aspettative.....	25
4.1.3 Descrizione.....	26
4.1.4 Ruoli di progetto .....	26
4.1.4.1 Responsabile di Progetto .....	26
4.1.4.2 Amministratore di Progetto .....	26

4.1.4.3 Analista .....	27
4.1.4.4 Progettista .....	27
4.1.4.5 Verificatore .....	27
4.1.4.6 Programmatore .....	27
4.1.5 Procedure .....	27
4.1.5.1 Gestione delle comunicazioni .....	27
4.1.5.1.1 Comunicazioni interne .....	28
4.1.5.1.2 Comunicazioni esterne .....	28
4.1.5.2 Gestione degli incontri .....	28
4.1.6 Rotazioni dei ruoli .....	29
4.1.7 Gestione dei compiti e task .....	29
4.1.7.1 Metodo organizzativo .....	30
4.1.7.1.1 Sprint Planning .....	31
4.1.7.1.2 Sprint Review .....	31
4.2 Infrastruttura .....	31
4.2.1 Strumenti .....	31
4.3 Miglioramento .....	32
4.4 Formazione .....	33
<b>A Standard ISO/IEC 12207 .....</b>	<b>34</b>
A.1 Processi primari .....	34
A.1.1 Acquisizione .....	34
A.1.2 Fornitura .....	34
A.1.3 Sviluppo .....	35
A.1.4 Operation .....	35
A.1.5 Manutenzione .....	35
A.2 Processi di supporto .....	36
A.2.1 Documentazione .....	36
A.2.2 Gestione della configurazione .....	36
A.2.3 Gestione della qualità .....	36
A.2.4 Verifica .....	36
A.2.5 Validazione .....	36
A.2.6 Revisione congiunta .....	36
A.2.7 Audit .....	36
A.2.8 Risoluzione dei problemi .....	37
A.3 Processi organizzativi .....	37
A.3.1 Gestione organizzativa .....	37
A.3.2 Gestione delle infrastrutture .....	37
A.3.3 Miglioramento .....	37
A.3.4 Formazione .....	37
<b>B Standard ISO/IEC 9126 .....</b>	<b>38</b>
B.1 Modello di qualità .....	38
B.1.1 Funzionalità .....	38

B.1.2 Affidabilità.....	38
B.1.3 Efficienza.....	39
B.1.4 Usabilità .....	39
B.1.5 Manutenibilità .....	39
B.1.6 Portabilità .....	40
B.2 Metriche per la qualità esterna.....	40
B.3 Metriche per la qualità interna .....	40
B.4 Metriche per la qualità in uso .....	40

# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento si pone lo scopo di individuare le best practices di progetto e definire correttamente il *Way of Working*<sup>G</sup> dell'attività produttiva, individuando facilmente una scansione del lavoro omogenea e coesa. Si riportano, al fine di garantire uno storico e un approccio incrementale nella sua realizzazione, le versioni e lo storico del documento. Qui vengono precisamente le convenzioni sull'uso degli strumenti utilizzati nella realizzazione dei processi di lavoro interno ed esterno, regolamentato da appositi standard<sup>G</sup> presenti in appendice a questo documento.

## 1.2 Scopo del prodotto

L'obiettivo del progetto è realizzare una piattaforma per la gestione delle traduzioni tramite un'infrastruttura multi-tenant<sup>G</sup>, permettendo di gestire testi in lingue diverse e a diversi utenti. Esso si rivolge a siti che offrono la possibilità di essere visualizzati in lingue differenti e gestire bilateralmente le traduzioni dei testi che li compongono, secondo un meccanismo di accesso, collaborazione, approvazione e rifiuto.

La piattaforma consente agli utenti di accedere al proprio Tenant e interagire a diverso livello con le traduzioni presenti. L'utilizzo delle tecnologie AWS<sup>G</sup> permette di riconoscere il tipo di utente tramite uno specifico token<sup>G</sup> a lui assegnato, permettendogli una specifica gestione della piattaforma. In particolare, possiamo distinguere:

- gli utenti finali (definiti come User<sup>G</sup>) potranno visualizzare un insieme di testi che dovranno tradurre;
- gli utenti amministratori (definiti come Admin<sup>G</sup>), che gestiscono un singolo Tenant, potranno impostare una serie di lingue secondarie, visualizzare e modificare testi originali, approvare e rifiutare le traduzioni presenti. Essi possono inoltre creare, modificare e cancellare categorie di traduzioni presenti e inviare testi tradotti assegnati;
- gli utenti definiti come SuperAdmin<sup>G</sup>, con permessi di gestione di tutti i Tenant, possono gestire ciascuno di questi e gli utenti ad essi associati.

Ogni traduzione è raggruppata in una categoria, e le traduzioni sono suddivise tra lingua di default impostata e una serie di lingue secondarie. Inoltre, l'applicativo consente di visualizzare le traduzioni eseguite o non eseguite e di visualizzare i testi tradotti in una lingua tramite API.

Per fornire la massima compatibilità, essa sarà fruibile tramite browser<sup>G</sup>, in grado di supportare correttamente le tecnologie di base HTML<sup>G</sup>, CSS<sup>G</sup>, JavaScript<sup>G</sup> ed altre successivamente definite.

## 1.3 Glossario

Al fine di evitare incomprensioni relative alla terminologia usata all'interno del documento, viene fornito un Glossario nel file apposito (Glossario v.2.0.0), tale da non avere terminologie ambigue nell'attività progettuale individuata e dandone una definizione precisa. Ogni termine avrà nel



documento una lettera G come apice, per meglio evidenziare la loro appartenenza al documento indicato.

## 1.4 Maturità e miglioramenti

Il presente documento è redatto con un approccio incrementale, al fine di poter implementare facilmente cambiamenti nel corso del tempo a seconda di esigenze concordate bilateralmente tra membri del gruppo e proponente. Pertanto, non può essere considerato definitivo e completo in questa versione.

## 1.5 Riferimenti

### 1.5.1 Riferimenti normativi

- Capitolato<sup>G</sup> C4-Piattaforma di localizzazione testi:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Progetto/C4.pdf>

### 1.5.2 Riferimenti informativi

- Slide T3 - Corso di Ingegneria del Software – Il ciclo di vita del SW:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T03.pdf>
- Slide T09 - Progettazione Software:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T07.pdf>
- Slide T14 - Verifica e Validazione: introduzione:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T10.pdf>
- Slide FC1 - Amministrazione di progetto:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/FC2.pdf>
- Slide T14 - Verifica e Validazione: analisi statica:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T11.pdf>
- Slide T14 - Verifica e Validazione: analisi dinamica:
  - <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T12.pdf>
- Standard ISO/IEC 12207
  - [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/IEC 9126
  - [http://www.colonese.it/00-Manuali\\_Pubblicatii/07-ISO-IEC9126\\_v2.pdf](http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf)
- Agile Manifesto
  - <https://agilemanifesto.org/iso/it/manifesto.html>

## 2 Processi Primari

### 2.1 Fornitura

#### 2.1.1 Descrizione, scopo e aspettative

Il processo di fornitura ha lo scopo di risolvere ogni dubbio legato alla volontà del proponente, al fine di evitare incomprensioni durante la sua realizzazione. Il gruppo *SWEG* intende mantenere dei rapporti di collaborazione fitti e riconosciuti con il proponente durante l'intero svolgimento del progetto, al fine di poter:

- determinare in modo univoco i needs soddisfatti dal prodotto finale;
- ottenere feedback in merito alle tecnologie utilizzate in fase progettuale e chiarimenti relativi a dubbi/difficoltà riscontrate;
- ottenere incontri di formazione sulle tecnologie consigliate e definire i vincoli per il prodotto finale, al fine di consolidare delle baseline solide di sviluppo del prodotto finale.

#### 2.1.2 Valutazione dei capitolati

Per poter determinare la corretta valutazione dei capitolati<sup>G</sup> proposti, il gruppo guidato dal *Responsabile* ha organizzato numerosi incontri interni, come testimoniato da relativi Verbali Interni ed Esterni contenenti i punti salienti di tali incontri. In questa prima fase, i vari membri hanno alternato il ruolo di *Analisti*, carpando per ogni capitolato:

- informazioni generali e dominio applicativo/tecnologico;
- valutazione oggettiva (pro e contro), completa disamina dei punti di forza e fattori critici;
- relative conclusioni in merito alle scelte operate.

Tale analisi può trovare riscontro nel relativo documento *Valutazione dei capitolati* a cui si è giunti per la scelta definitiva.

#### 2.1.3 Piano di Progetto

Il *Piano di Progetto* è situato nella prima fase di pianificazione e costituisce uno strumento di pianificazione che definisce tutte le attività che dovranno essere svolte, le risorse necessarie e la scadenza del progetto stesso. E' un documento ufficiale, quindi soggetto a versionamento e ad approvazione, e viene utilizzato per descrivere in modo chiaro e conciso gli obiettivi di progetto e gli elementi necessari per il loro compimento.

Nello specifico, un *Piano di Progetto* è caratterizzato da:

- **Analisi dei rischi:** analisi delle difficoltà che potrebbero crearsi durante lo sviluppo del progetto ed evitare problemi che possano causare rallentamenti e che possono ostacolare l'attività di progetto. Questi rischi possono essere classificati in due tipi:
  - Rischi organizzativi e personali;

- Rischi tecnologici.
- **Modello di sviluppo:** il gruppo decide che modello utilizzare;
- **Pianificazione:** viene scelta una scansione degli eventi per le attività di progetto da svolgere, definendo i periodi;
- **Preventivo:** riepilogo schematico del prospetto economico e prospetto orario totale necessario per il progetto e inoltre riepilogo diviso per ogni fase;
- **Consuntivo:** traccia dell'andamento rispetto al preventivo fatto.

#### 2.1.4 Piano di Qualifica

Il documento rappresenta i compiti e le attività relative al progetto, che dovranno essere svolte dal *Verificatore*, necessarie a garantire la qualità del prodotto finale utilizzando tecniche di verifica e di validazione<sup>G</sup>.

Più in specifico, il *Piano di Qualifica* è formato da:

- **Qualità di processo:** scelta delle metriche per il controllo della qualità di processo;
- **Qualità di prodotto:** scelta delle metriche per il controllo della qualità di prodotto;
- **Test e specifiche:** specifica dei test che verranno eseguiti sul prodotto per garantire i requisiti stabiliti;
- **Resoconto e valutazioni per miglioramento:** risultati dell'attività di verifica svolta.

#### 2.1.5 Collaudo e rilascio

La fase di collaudo serve al gruppo per verificare che i requisiti definiti in fase di analisi vengano rispettati e non si verifichino errori. Se il collaudatore riscontra anomalie dovranno essere risolte, invece se il prodotto non presenta errori, il *Responsabile di Progetto* presenterà al committente il consuntivo finale e verrà consegnato codice sorgente e la documentazione relativa al progetto.

Infine, dopo che il prodotto è stato rilasciato, non è prevista nessuna fase di manutenzione<sup>G</sup>.

## 2.2 Sviluppo

### 2.2.1 Descrizione, scopo e aspettative

Lo scopo del processo di sviluppo è definire i compiti e le attività che il gruppo deve svolgere per la codifica del prodotto software richiesto, definendo dei ruoli<sup>G</sup>.

Per produrre un software che rispetti le aspettative del proponente il gruppo *SWEG* determina gli obiettivi di sviluppo e design da raggiungere, determinando la corretta implementazione del prodotto finale affinché sia:

- conforme alle richieste del proponente, come dettagliato dall'Analisi dei Requisiti e realizzato dal design ER;
- conforme agli obiettivi di sviluppo e superi i test di verifica<sup>G</sup> e validazione<sup>G</sup>.

Nello specifico, sono dettagliate le seguenti attività:

## 2.2.2 Analisi dei requisiti

L'analisi dei requisiti è l'attività che si compie prima di poter chiaramente definire il prodotto finale al fine di determinare univocamente, grazie all'utilizzo di Analisti, informazioni maggiori e capillari del dominio d'uso e dello scopo del prodotto finale. In particolare, possiamo individuare:

- lo scopo del prodotto finale, andando incontro alle aspettative ed obiettivi del proponente;
- definizione di funzionalità del prodotto finale, individuando figure che interagiscono con il sistema (attori) e relative interazioni (casi d'uso);
- definizione della qualifica e dell'attività di controllo dei test;
- confronti interni ed esterni al gruppo, definendo una visione di massima del prodotto e della relativa specifica, quantificando gli impegni sulla base dei ruoli individuati e relativa rotazione.

Al fine di poter fornire una corretta interpretazione e realizzazione del prodotto, l'attività ha lo scopo di comprendere la specifica del capitolato, sulla base di un confronto mirato con il proponente, interpretando e ampliando la relativa realizzazione.

### 2.2.2.1 Denominazione e legenda

#### 2.2.2.1.1 Struttura Casi d'uso

I casi d'uso hanno una struttura ben definita ed è come segue:

- Attori;
- Precondizioni;
- Postcondizioni;
- Scenario principale;
- Estensioni (se esistono).

#### 2.2.2.1.2 Denominazione Casi d'uso

Descrive come l'utente utilizza un sistema per raggiungere un determinato obiettivo. Ogni caso d'uso è identificato univocamente secondo lo schema seguente:

**UC[Num].[Sottocaso]-[Titolo]**

dove:

- Num: Numero del caso d'uso in questione;
- Sottocaso: sottocaso del caso d'uso in questione;
- Titolo: breve titolo che descrive il caso d'uso.

#### 2.2.2.1.3 Struttura Requisiti

I requisiti individuati nell'*Analisi dei Requisiti* sono stati strutturati nel modo seguente:

- Codice;
- Descrizione;
- Classificazione: può assumere valori diversi:

- Requisito obbligatorio;
- Requisito facoltativo;
- Requisito desiderabile (ma non obbligatorio).
- Fonti.

#### 2.2.2.1.4 Denominazione Requisiti

Ogni requisito è identificato univocamente secondo lo schema seguente:

**R[Tipo].[Codice]**

dove:

- Tipo: indica la tipologia del Requisito e può assumere i seguenti valori:
  - F: Requisiti funzionali;
  - Q: Requisiti di qualità;
  - S: Requisiti di sistema;
  - P: Requisiti prestazionali e di sicurezza.
- Codice: numero identificativo incrementale e univoco in base alla tipologia. Il codice può presentare dei sottocasi che verranno strutturati aggiungendo un “.” con un numero che indicherà il sottocaso in questione. Un esempio è il seguente:

**R[Tipo].[Codice].[SottoCodice]**

### 2.2.3 Progettazione

#### 2.2.3.1 Descrizione, scopo e aspettative

La fase di progettazione è essenziale per definire le linee essenziali della struttura del progetto in funzione dei requisiti evidenziati nell’analisi e specificati nell’Analisi dei Requisiti. Questa attività è svolta dai progettisti che si occupano di creare un piano per implementare tutti i requisiti specificati.

Il nostro gruppo, per questa fase del ciclo di vita del software, ha le seguenti aspettative:

- Trasformazione di tutti i requisiti in specifiche dettagliate che coprono tutti gli aspetti del sistema;
- Facile comprensione per garantire una manutenzione facilitata;
- Approvazione del passaggio alla fase di sviluppo.

Questo processo può essere suddiviso nei seguenti tre livelli:

- **Design dell’interfaccia:** questa fase procede ad un alto livello di astrazione rispetto al funzionamento interno del sistema, cioè, durante la progettazione dell’interfaccia, l’attenzione è focalizzata sulle tecnologie che verranno utilizzate nella fase di sviluppo del software. Questa particolare livello porta alla produzione di un Proof of Concept;
- **Progettazione architettonica:** in questa fase viene scelta la struttura generale del sistema, facendo una definizione ad alto livello ignorando i dettagli interni dei componenti principali del prodotto. Inoltre si definiscono i test di integrazione;

- **Progettazione dettagliata:** specifica degli elementi interni di tutti i principali componenti e specifiche architetture del prodotto. Inoltre si definiscono i diagrammi delle classi e i test di unità per ogni componente. Questa fase della progettazione costituisce la *Product Baseline*.

## 2.2.4 Codifica

### 2.2.4.1 Descrizione, scopo e aspettative

Dopo la fase di progettazione del prodotto, i membri del gruppo con il ruolo di *Programmatore* passeranno alla fase di codifica implementando le specifiche dei requisiti e i documenti di progettazione.

Lo scopo di questa fase è concretizzare il tutto realizzando il prodotto software desiderato attraverso l'attività di programmazione.

Le aspettative per la fase di codifica sono:

- completare lo sviluppo del prodotto finale garantendo la qualità di esso e rispettando le richieste del proponente;
- assicurare che il codice prodotto sia facilmente leggibile.

# 3 Processi di Supporto

## 3.1 Documentazione

### 3.1.1 Descrizione, scopo e aspettative

La documentazione software è l'informazione rappresentata in testo scritto che accompagna un software, utile a descrivere il prodotto alle persone che lo sviluppano, lo distribuiscono e lo utilizzano. Intende infatti facilitare i membri di un team durante lo sviluppo, tenendo traccia e documentando tutti i processi e attività di sviluppo presenti andando a facilitare anche la manutenzione migliorando la qualità del risultato finale.

Da come si evince dalla descrizione, la documentazione software ricopre un ruolo molto importante nel processo di sviluppo di un software e le aspettative per essa sono:

- avere delle regole chiare e concise utili per la stesura di un documento;
- una struttura chiara e comune per tutti i documenti, nel ciclo di vita del software per garantire una omogeneità.

### 3.1.2 Ciclo di vita

Il ciclo di vita di un documento è composto da tre fasi:

- Redazione: fase di stesura del documento divisa tra i membri del gruppo utilizzando il versionamento.
- Verifica: il documento dopo essere stato scritto affronta un processo di verifica. Essa può avvenire per parti del documento oppure dopo la stesura totale. Quindi, ogni parte dovrà essere verificata da un verificatore (una persona o più) che dovrà essere una persona diversa dal Redattore di quella specifica parte;
- Approvazione: il Responsabile di progetto, una volta che il documento è interamente completato e verificato, lo approva.

### 3.1.3 Template

Il gruppo *SWEG* ha deciso di adottare un semplice Template standardizzato sviluppato utilizzando Word. L'uso di questo template è standardizzato per tutti i documenti.

### 3.1.4 Documenti prodotti

I documenti prodotti possono essere classificati in due categorie principali, presentate in queste sottosezioni.

#### 3.1.4.1 Formali

Sono documenti con nome univoco utilizzati per regolare le attività all'interno del gruppo durante tutto il ciclo di vita del software. Sono storicizzati utilizzando il versionamento e approvati da parte del *Responsabile di Progetto*.

Si dividono in due categorie: ad uso interno e ad uso esterno.

I primi sono destinati all'utilizzo da parte dei soli membri del gruppo:

- Norme di Progetto;
- verbali, a loro volta divisi tra:
  - interni, scritti in base agli incontri tra i membri del gruppo;
  - esterni, scritti sulla base degli incontri asincroni con il proponente.

I secondi invece sono destinati all'utilizzo da parte di persone esterne al gruppo, infatti interesseranno sia al committente che al proponente, consegnati nell'ultima versione approvata. Sono definiti documenti esterni:

- Analisi dei Requisiti;
- Specifica Tecnica;
- Specifica Architettuale
- Manuale Utente;
- Manuale Sviluppatore;

- Piano di Progetto;
- Piano di Qualifica;
- Glossario.

### 3.1.4.2 Informali

Categoria di documenti dove sono presenti:

- Documenti che non ancora stati approvati dal responsabile del progetto;
- Stesura di brevi bozze;
- Documenti che non necessitano di versionamento.

Questi particolari documenti sono gestiti in una sezione divisa, infatti il nostro gruppo ha preferito predisporre un Google Drive condiviso dove poterli gestire con facilità.

### 3.1.5 Struttura del documento

I documenti ufficiali hanno una struttura precisa e comune che deve essere rigorosamente rispettata.

#### 3.1.5.1 Prima pagina

Nella prima pagina troviamo:

- il titolo che rappresenta il nome del nostro gruppo SWEG, accompagnato subito al di sotto da un titolo minore dove l'acronimo viene messo per esteso;
- la mail del gruppo, cioè [swegsoftware@gmail.com](mailto:swegsoftware@gmail.com);
- nome del documento;
- informazioni sul documento dove sono scritti i vari nomi dei redattori, verificatori, l'amministratore, i destinatari e il numero della versione.

#### 3.1.5.2 Registro dei Cambiamenti - Changelog

Il registro dei cambiamenti è utile per tracciare la storicità del documento. Nella tabella sono riportati:

- La versione del documento dove sono presenti tutti i versionamenti che sono stati eseguiti durante la fase di stesura;
- La data della modifica al documento;
- L'autore che ha eseguito il cambiamento;
- Il verificatore che verifica le modifiche;
- La descrizione dove vengono riportate le modifiche o aggiunte fatte rispetto alla versione precedente.



### 3.1.5.3 Indice

Tutti i documenti, ad eccezione dei verbali che hanno una struttura a parte, hanno un indice nella pagina seguente al registro dei cambiamenti. Essi servono per migliorare la navigazione all'interno del documento, agevolando la lettura e la ricerca di specifiche sezioni. Ogni indice parte dal numero "1".

### 3.1.5.4 Piè di pagina

In ogni pagina è presente al di sotto una sezione chiamata piè di pagina composta dal logo *SWEG* sia nel formato sigla che subito accanto nel formato esteso. A questo si accompagna il nome del documento.

### 3.1.5.5 Verbali

Questo particolare tipo di documento, è un'eccezione perché ha delle variazioni rispetto alla struttura illustrata sopra. Infatti, non è presente la tabella dei cambiamenti che non avrebbe un utilizzo, e l'indice vista la brevità del documento. La struttura dopo la prima pagina, che rimane invariata, si presenta in questo modo:

- Partecipanti: in questa sezione troviamo per prima cosa l'orario di inizio e di fine dell'incontro. Subito sotto è presente una tabella con la prima colonna per il nome del partecipante e la seconda che indica la durata della presenza di ogni partecipante;
- Sintesi ed elaborazione incontro: parte dove viene scritto un riassunto degli argomenti trattati durante l'incontro, con una sezione finale per gli eventuali dubbi emersi o indicazioni per il prossimo incontro che si svolgerà.

Inoltre i verbali saranno divisi a seconda se la riunione è interna, quindi tra i membri del gruppo *SWEG* oppure esterna, per esempio con l'azienda.

## 3.1.6 Norme tipografiche

### 3.1.6.1 Nome del file

I nomi dei file hanno tutti una notazione omogenea tra di loro, dove la lettera iniziale è sempre maiuscola e il resto tutto minuscolo. Viene fatta eccezione per i verbali dove il titolo è composto semplicemente dalla data in cui si è svolto l'incontro. La data viene scritta in formato AAAA-MM-DD.

### 3.1.6.2 Stile del testo

Gli stili del testo che vengono apportati all'interno di tutti i documenti sono:

- Grassetto: viene utilizzato per i titoli delle sezioni ed è riservato a parole con una certa importanza.
- Sottolineato: utilizzato solamente per i link presenti nei documenti.
- Monospazio: utilizzato per riferimenti a codice.

### 3.1.6.3 Glossario

Il Glossario è un documento dove è presente una raccolta di vocaboli che il gruppo ritiene abbiano bisogno di una spiegazione per comprendere al meglio il giusto significato evitando ambiguità.

I termini sono registrati in ordine alfabetico e seguiti dalla dichiarazione del significato o accompagnati da altre osservazioni.

Ciascun termine che sarà presente nel Glossario avrà una G come apice all'interno dei documenti, per rendere il termine subito riconoscibile. Generalmente, l'apice è presente solo per la prima volta che si incontra la parola, per le successive conteggiamo la parola come un termine non più ambiguo ma già conosciuto. Si escludono dal Glossario titoli o sottotitoli e didascalie di immagini o tabelle.

Il documento in questione sarà accessibile a tutti i membri del gruppo, e conservato all'interno del Google Drive dove sarà più facile poter aggiungere possibili termini con un particolare significato durante la stesura dei documenti. Quindi qualsiasi membro del gruppo dopo aver aggiunto la parola e la relativa definizione, dovrà verificare che il termine non fosse già presente all'interno del Glossario, evitando così possibili doppioni.

### 3.1.6.4 Altre Norme Tipografiche

- I nomi dei documenti sono sempre scritti in formato esteso all'interno dei documenti;
- i nomi dei ruoli vengono sempre scritti in formato esteso all'interno dei documenti;
- i nomi delle sezioni e sottosezioni iniziano sempre con la lettera maiuscola, e ogni parola successiva ha sempre la prima lettera in carattere maiuscolo. Queste possiedono indici cliccabili e di riferimento;
- ogni contenuto grafico (tabelle, immagini o grafici stessi) sono indicizzati con apposito indice presente ad inizio documento dopo il Sommario;
- qualsiasi elenco inizia sempre con la lettera maiuscola;
- gli elementi all'interno di un elenco, finiscono con il carattere “;”, a meno che non sia l'ultimo elemento dell'elenco, dove il carattere finale sarà un “.”.

## 3.1.7 Elementi grafici

### 3.1.7.1 Tabelle

Le tabelle presenti all'interno dei documenti ufficiali sono contrassegnate da un titolo iniziale che descrive il contenuto della tabella e da una didascalia descrittiva del contenuto, posta sotto di essa e centrata rispetto alla pagina.

Ogni didascalia viene identificata da l'identificativo con sintassi:

**Tabella [n]: Descrizione**

dove:

- [n] rappresenta il numero assoluto della tabella all'interno del documento;
- Descrizione, contiene una breve descrizione del contenuto della tabella.

Successivamente le righe delle tabelle sono alternate con colori diversi per permettere una semplificazione nella lettura e infine la tabella deve essere centrata orizzontalmente rispetto alla pagina.

Eccezione viene fatta per i verbali dove la tabella non ha né la didascalia né è centrata orizzontalmente rispetto al documento.

Le tabelle delle modifiche presenti in ogni documento e le tabelle presenti nei verbali, sono eccezioni in quanto non seguono le regole appena descritte, infatti non hanno nessuna didascalia.

### **3.1.7.2 Immagini**

Le immagini, all'interno dei documenti devono essere centrate orizzontalmente rispetto alla pagina.

Ogni immagine all'interno dei documenti è contrassegnata da una didascalia descrittiva di ciò che l'immagine rappresenta. La didascalia è posta al di sotto dell'immagine, è centrata orizzontalmente rispetto alla pagina e ha il seguente sintassi:

#### **Immagine [n]: Descrizione**

dove:

- [n] rappresenta il numero assoluto dell'immagine all'interno del documento;
- Descrizione, contiene una breve descrizione del contenuto dell'immagine.

Eventuali aggiunte di diagrammi uml all'interno dei documenti verranno inseriti nei documenti come immagini.

### **3.1.7.3 Grafici**

I grafici, all'interno dei documenti devono essere centrati all'interno della pagina orizzontalmente e ognuno deve essere contrassegnato da una didascalia descrittiva di ciò che rappresenta.

La didascalia è posta al di sotto del grafico, è centrata orizzontalmente rispetto alla pagina e ha la seguente sintassi:

#### **Grafico[n]: Descrizione**

dove:

- [n] rappresenta il numero assoluto del grafico all'interno del documento;
- Descrizione, contiene una breve descrizione del contenuto del grafico.

### 3.1.8 Strumenti

#### 3.1.8.1 Microsoft Word

Per la stesura dei documenti ufficiali viene utilizzato *Word*, dove vengono seguite precise norme tipografiche a livello di indicizzazione dei contenuti e dei capitoli.

#### 3.1.8.2 Google Docs

Per la stesura delle bozze dei documenti o brevi note viene utilizzato *Google Docs*, preferito dal gruppo per avere sia un accesso ai documenti più facile e veloce, che per non avere problemi di più modifiche simultanee nello stesso documento.

#### 3.1.8.3 UML

Per la creazione, sviluppo e salvataggio dei diagrammi UML<sup>6</sup> il nostro gruppo ha scelto di affidarsi al software *StarUML*.

## 3.2 Gestione della Configurazione

### 3.2.1 Descrizione, scopo e aspettative

La gestione della configurazione è il processo utilizzato per mantenere software in un determinato stato coerente, con il fine di assicurare che il sistema funziona nel modo previsto nonostante modifiche che potrebbero venir apportate durante il tempo.

Possibili problemi nella configurazione potrebbero essere la causa di incoerenza o problemi di conformità che influenzano negativamente le operazioni.

Il gruppo *SWEG* si aspetta, attraverso la gestione della configurazione di:

- Risolvere problemi prima che diventino insormontabili;
- Semplificare il tracciamento delle modifiche e l'individuazione di errori.

### 3.2.2 Versionamento

Il versionamento consente di tracciare i cambiamenti occorsi ad un documento. Permette, inoltre, di riportare un documento ad uno stadio precedente e consente di visualizzare le modifiche effettuate nel corso del tempo con il relativo autore.

Il nostro gruppo ha deciso di utilizzare il seguente codice per rappresentare la versione di un documento:

[x].[y].[z]

Dove:

- x: è un numero intero che parte da 0 che viene incrementato dal RdP dopo la sua approvazione (versione di produzione);
- y: è un numero intero incrementato dal Ve ad ogni verifica;

- z: è un numero intero incrementato dal Re ad ogni piccola modifica.

### 3.2.3 Repository

#### 3.2.3.1 Tecnologie

Il nostro gruppo ha deciso di utilizzare per la gestione della configurazione il servizio *GitHub*<sup>G</sup>, basato sul sistema di controllo di versione distribuito *Git*<sup>G</sup>.

La repository è pubblica e si può facilmente trovare al seguente link:

<https://github.com/SWEG-Software-Engineering-Group/docs>

Inoltre, per consentire una più semplice navigazione e consultazione degli stessi, il gruppo ha predisposto un sito apposito, tramite GitHub, con il seguente riferimento:

<https://sweg-software-engineering-group.github.io/>

#### 3.2.3.2 Struttura repository

La struttura si presenta con un unico branch, nel quale sono presenti le cartelle:

- Verbali, contenente tutti i verbali interni ed esterni;
- la cartella “1 - Candidatura”, comprensiva di lettera di presentazione, il preventivo costi e assunzione impegni, la valutazione capitolati e i verbali;
- la cartella “2 - RTB”, dove sono presenti:
  - la cartella “Documentazione Esterna”, comprensiva di Analisi dei Requisiti, Glossario, Piano di Progetto e Piano di Qualifica;
  - la cartella “Documentazione Interna” con all’interno il documento Norme di Progetto e i relativi verbali, suddivisi a loro volta come specificato tra interni ed esterni.
- la cartella “3 - PB”, dove sono presenti:
  - la cartella “Documentazione Esterna”, comprensiva di Analisi dei Requisiti, Glossario, Piano di Progetto, Manuale Utente, Manuale Sviluppatore, Specifica Tecnica e Piano di Qualifica;
  - la cartella “Documentazione Interna” con all’interno il documento Norme di Progetto e i relativi verbali, con ulteriore suddivisione sopra specificata.

Per il Proof of Concept, è stato creato un repository separato, contenente tutti i file sorgente relativi al PoC<sup>G</sup> che sarà consegnato in occasione della prima revisione (RTB).

In questi rami troveremo solo file con estensione *.pdf* e *.png*. Infine nel main troveremo la cartella “.github” che contiene i file generati automaticamente durante la compilazione.

## 3.3 Gestione della qualità

### 3.3.1 Descrizione, scopo e aspettative

La gestione della qualità di progetto è l'insieme dei processi e delle attività che vengono eseguite all'interno di un progetto per garantire che la qualità dei *deliverable*<sup>G</sup> e delle performance del progetto siano in accordo con gli obiettivi e i requisiti del progetto stesso.

I membri del nostro gruppo hanno come obiettivi:

- Comprendere, valutare e gestire le aspettative affinché i requisiti del richiedente vengano rispettati;
- Attraverso questo processo riuscire ad impostare chiaramente i requisiti di qualità e documentare tutte le procedure necessarie per completare il progetto secondo le aspettative richieste;
- Consegnare il progetto seguendo il piano di qualità, così da consegnare il prodotto in tempo, rispettando il budget e secondo i requisiti e aspettative del proponente.

### 3.3.2 Piano di Qualifica

Per rispettare tutti i punti di questo processo utilizziamo il Piano di Qualifica, un documento dove abbiamo predisposto un elenco dei dati da inserire per avere un piano di alta qualità:

- definire i requisiti del richiedente;
- definire metriche/parametri per analizzare i dati;
- definire un sistema per il controllo delle qualità durante tutto il ciclo di vita del progetto;
- definire un piano di miglioramento che descriva le azioni per analizzare le prestazioni di qualità e identificare le attività per migliorare il valore del progetto.

### 3.3.3 Denominazione metriche

Le metriche utilizzate vengono definite di seguito:

**M[Tipologia][Acronimo]**

Dove:

- **Tipologia** indica il tipo della metrica:
  - PC: per processo;
  - PD: per prodotto;
- **Acronimo** del nome della metrica utilizzata.

## 3.4 Verifica

### 3.4.1 Descrizione, scopo e aspettative

La verifica del software è definita come il processo di valutazione del prodotto software, per garantire che la fase di sviluppo venga eseguita in modo accurato, per costruire il prodotto software desiderato. Viene eseguito durante la fase in corso di sviluppo del software, per garantire il rilevamento di difetti e guasti nella fase iniziale del ciclo di vita dello sviluppo e per determinare se soddisfa i requisiti del cliente.

Le aspettative per questo processo sono:

- Aumentare la fiducia del gruppo nell'avanzare nello sviluppo del progetto in modo corretto;
- Ottenere il prodotto finale atteso;
- Individuazione precoce di errori che possono rivelarsi un risparmio sia in costi che in tempo.

### 3.4.2 Tipi di Verifica

Il processo di verifica è formato da due tipi, ognuno dei quali è focalizzato sulla verifica di vari aspetti del software. Insieme, questi due tipi di verifica assicurano che il software sia conforme e soddisfi i requisiti specificati.

I due tipi verranno elencati nei due punti seguenti, aggiungendone uno finale per la verifica della documentazione.

#### 3.4.2.1 Analisi statica

L'analisi statica comporta l'ispezione del codice prima della sua esecuzione e garantisce che il software soddisfi i requisiti e le specifiche specificati. Non essendo richiesta nessuna esecuzione dell'oggetto in verifica, questo tipo particolare di analisi non è applicabile solo al codice, ma anche ai documenti.

E' un approccio effettuato semplicemente facendo un'analisi degli aspetti statici del sistema software, come le convenzioni del codice o il calcolo di metriche per il software. Questa procedura, inoltre, comprende sia tecniche di test manuali che automatizzate, come tecniche per la coerenza.

Questa tipologia particolare di analisi si divide in due metodi possibili:

- **Walkthroughs:** attraverso una lettura di ampio spettro si va ad esaminare e successivamente discutere su possibili errori o difetti trovati. Metodo utile nel caso non si abbia la certezza sulla posizione delle possibili problematiche;
- **Inspection:** è un metodo molto efficace per rimuovere errori e difetti, utilizzando un approccio mirato ad essi. Ciò significa che si ha già presente un'idea sul come capire cosa potrebbe creare eventuali problemi.

### 3.4.2.2 Analisi Dinamica

L'analisi dinamica viene eseguita insieme all'esecuzione del software e consiste semplicemente nella fase di test.

A differenza della verifica statica, quella dinamica comporta l'esecuzione del sistema e dei suoi componenti.

Il gruppo, inoltre utilizzerà un insieme di test che dovranno essere ripetibili ed automatizzati, nel caso del gruppo *SWEG* affidandosi alla libreria di test di React<sup>G</sup> e lo strumento Jest per la parte front-end<sup>G</sup>, il citato strumento Jest e GitHub Actions<sup>G</sup> per la parte back-end<sup>G</sup>.

### 3.4.2.3 Denominazione dei test

Le specifiche dei test sono dettagliate nel documento *Piano di Qualifica*, a cui si rimanda per il dettaglio e l'effettiva classificazione dei tipi di test con relativo dettaglio implementativo. La descrizione di massima del documento a livello normativo è documentata in una sezione precedente.

Al fine di garantire di creare una denominazione uniforme e per facilitare la comprensione, i test sono identificati da un codice come segue:

**T[Tipologia][Identificativo]**

Dove:

- **Tipologia** indica il tipo del test eseguito:
  - U: per Test di Unità;
  - I: per Test di Integrazione;
  - S: per Test di Sistema;
  - A: per Test di Accettazione;
  - R: per Test di Regressione.
- **Identificativo** del test in oggetto

Ove previsto dal tipo di test, si dettaglia questa ulteriore denominazione interna, creata ai fini di facilitare il tracciamento delle attività dedicate:

**T[Tipologia][Identificativo Test]-[Identificativo Requisito]\*[Importanza Requisito] [IdNumerico]**

Dove:

- **Tipologia** indica il tipo del test eseguito:
  - U: per Test di Unità;
  - I: per Test di Integrazione;
  - S: per Test di Sistema;
  - A: per Test di Accettazione;
  - R: per Test di Regressione.
- **Identificativo** del test in oggetto
- **Identificativo Requisito** è un valore crescente che distingue i requisiti della stessa tipologia;
- **Importanza Requisito**, valore testuale che definisce l'importanza dei test in oggetto, come stabilito dalla sezione §4.1 del documento *Analisi dei Requisiti*;



- **Id Numerico** è un codice di riferimento crescente per ciascun test dello stesso tipo.

Grazie a questa denominazione, il gruppo rende immediata la comprensione del tipo di test relativo ad un certo requisito, non ritenendo necessario inserire un'ulteriore tabella di tracciamento, in quanto si rende chiaro il tracciamento biunivoco [test-requisito] nel mondo indicato.

#### 3.4.2.4 Verifica della Documentazione

La verifica della documentazione è formata da:

- Controllo ortografia e sintassi;
- Controllo dell'utilizzo nel modo corretto delle norme tipografiche o qualsiasi altra norma di stile o formattazione accordata;
- Verifica della pertinenza dei contenuti scritti.

## 3.5 Validazione

### 3.5.1 Descrizione, scopo e aspettative

La convalida del software è un processo di valutazione del prodotto, in modo da garantire che il software soddisfi i requisiti del richiedente predefiniti e specificati, nonché le richieste e le aspettative degli utenti finali/proponente. Quando c'è stata una buona verifica in tutta la fase di sviluppo, il processo di validazione risulta avere successo.

L'esito finale positivo della validazione garantisce che il prodotto finale è in linea rispetto alle aspettative.

Le aspettative per questo processo sono:

- Rilevare possibili errori ignorati o trascurati durante la fase di verifica;
- Soddisfare i requisiti specificati nell'Analisi dei Requisiti per il prodotto finale;
- Aiuta a migliorare la qualità e il valore del prodotto software finale.

## 4 Processi Organizzativi

### 4.1 Gestione dei processi

#### 4.1.1 Scopo

Lo scopo del processo è la creazione del documento denominato *Piano di Progetto*, utilizzato dai componenti del gruppo per l'organizzazione e la gestione dei ruoli di ogni componente.

#### 4.1.2 Aspettative

Le principali aspettative del processo organizzativo riguardante la gestione dei processi consistono nella:

- Redazione del documento Piano di Progetto;
- Definizione dei ruoli assunti dai membri del gruppo;
- Definizione di un piano per l'esecuzione dei compiti programmati.

### 4.1.3 Descrizione

Le attività previste dal processo organizzativo di gestione dei processi sono raccolte nel Piano di Progetto, la cui redazione è a carico del Responsabile di Progetto, con la collaborazione di uno o più Amministratori.

Vengono trattati nello specifico i seguenti argomenti:

- Ruoli all'interno del progetto;
- Comunicazioni;
- Incontri;
- Strumenti di coordinamento;
- Strumenti di versionamento;
- Rischi.

### 4.1.4 Ruoli di progetto

Il Responsabile di Progetto si occupa di suddividere i ruoli e l'assegnazione oraria per i membri del gruppo, garantendo che ognuno di essi assuma, nel corso del progetto, almeno una volta ogni ruolo. I ruoli richiesti dal progetto sono qui di seguito descritti.

#### 4.1.4.1 Responsabile di Progetto

Figura professionale, punto di riferimento sia per il committente sia per il fornitore, con lo scopo di mediare tra le due parti.

Si assume la responsabilità delle scelte del gruppo dopo averle approvate.

In particolare, si occupa di:

- approvare l'emissione della documentazione;
- approvare l'offerta economica sottoposta al committente;
- pianificare e coordinare le attività di progetto;
- gestire le risorse umane;
- studiare e gestire i rischi.

#### 4.1.4.2 Amministratore di Progetto

Figura professionale con l'incarico delle procedure di controllo e amministrazione dell'ambiente di lavoro, con piena responsabilità sulla capacità operativa e sull'efficienza.

In particolare, si occupa di:

- ricercare, studiare e mettere in opera risorse per migliorare l'ambiente di lavoro e automatizzarlo ove possibile;
- risolvere problemi legati alla gestione dei processi;
- salvaguardare la documentazione di progetto;
- effettuare il controllo di versioni e configurazioni del prodotto software;
- redigere e attuare i piani e le procedure per la gestione della qualità.

#### 4.1.4.3 Analista

Figura professionale con maggiori competenze riguardo il dominio applicativo del problema.

Si occupa di:

- studiare il problema e il relativo contesto applicativo;
- comprendere il problema e definire la complessità e i requisiti;
- redigere l'Analisi dei Requisiti.

#### 4.1.4.4 Progettista

Figura professionale che gestisce gli aspetti tecnologici e tecnici del progetto.

In particolare si occupa di:

- effettuare scelte riguardanti gli aspetti tecnici e tecnologici del progetto, favorendone l'efficacia G e l'efficienza G;
- definire un'architettura del prodotto da sviluppare che miri all'economicità e alla manutenibilità a partire dal lavoro svolto dall'analista;
- redigere la *Specifica Tecnica* e la parte pragmatica del *Piano di Qualifica*.

#### 4.1.4.5 Verificatore

Figura professionale con l'incarico di sorveglianza sul lavoro svolto dagli altri componenti del gruppo, sulla base delle proprie competenze tecniche, esperienza e conoscenza delle norme.

In particolare si occupa di:

- Esaminare i prodotti in fase di revisione, con l'ausilio delle tecniche e degli strumenti definiti nelle *Norme di Progetto*;
- Verificare la conformità dei prodotti ai requisiti funzionali e di qualità;
- Segnalare eventuali errori.

#### 4.1.4.6 Programmatore

Figura professionale incaricata alla codifica del progetto e delle componenti di supporto che verranno utilizzate per eseguire prove di verifica e validazione sul prodotto.

In particolare, si occupa di:

- implementare la *Specifica Tecnica* redatta dal Progettista;
- scrivere un codice pulito e facile da mantenere che rispetti le *Norme di Progetto*;
- realizzare gli strumenti per la verifica e la validazione del software;
- redigere il *Manuale Utente* relativo alla propria codifica.

### 4.1.5 Procedure

#### 4.1.5.1 Gestione delle comunicazioni

Per il coordinamento e le comunicazioni durante l'intera durata del progetto il gruppo *SWEG* utilizzerà le seguenti procedure:

- **Comunicazione interna:** coinvolge tutti i membri del gruppo;
- **Comunicazione esterna:** coinvolge proponente e committenti.

#### 4.1.5.1.1 Comunicazioni interne

Le comunicazioni interne avvengono tramite le applicazioni:

- Telegram<sup>G</sup>: permette una comunicazione veloce tra i membri del gruppo e viene utilizzato principalmente per organizzare incontri interni o discutere su eventuali quesiti.
- Discord<sup>G</sup>: il nostro gruppo ha scelto questa applicazione per comodità e facilità di comunicazione attraverso canale vocale.

#### 4.1.5.1.2 Comunicazioni esterne

Le comunicazioni esterne vengono gestite dal *Responsabile di Progetto* tramite l'utilizzo dei seguenti canali:

- **Posta Elettronica:** tramite l'indirizzo del gruppo [swegsoftware@gmail.com](mailto:swegsoftware@gmail.com)
- **Slack<sup>G</sup>:** canale dato dall'azienda al gruppo SWEG per avere un canale diretto con il proponente.

#### 4.1.5.2 Gestione degli incontri

##### 4.1.5.2.1 Incontri interni

Negli incontri interni i partecipanti sono solo i membri che compongono il gruppo. Gli incontri interni solitamente vengono svolti almeno una volta a settimana, di Mercoledì (nei primi 4 periodi come indicato da Piano di Progetto) oppure di Venerdì (nei secondi 4 periodi) e in caso ne servisse un secondo, viene scelto un giorno che può comprendere anche il fine settimana.

Un incontro può essere richiesto da qualsiasi membro del gruppo al *Responsabile di Progetto* che poi organizzerà a seconda della disponibilità di tutti i membri. Dopo aver fissare definitivamente la data e l'ora dell'incontro, viene comunicato tramite il gruppo telegram.

La modalità di incontro che il nostro gruppo ha scelto è solamente virtuale e non fisica vista la difficoltà che ci sarebbero. Infatti, come precedentemente accennato, viene utilizzato Discord che permette una comunicazione sincrona vocale affidabile e comoda per tutti i membri del gruppo.

Per mantenere una buona efficienza e per ottimizzare al massimo il tempo limitato di ogni incontro interno, viene seguita una linea guida in ogni riunione sincrona:

- prima di ogni incontro, avere una scaletta dei principali punti da vedere durante la riunione;
- discussione sul lavoro che ogni membro del gruppo ha svolto dall'ultimo incontro;
- discussione di tutti i punti prefissati nella scaletta iniziale, con un confronto su eventuali dubbi;
- pianificazione delle attività da svolgere per ogni membro del gruppo, fino al prossimo incontro.

Alla fine di ogni incontro:

- il *Responsabile di Progetto* si occuperà di scrivere un breve resoconto sulle attività svolte durante la riunione sincrone, per permettere alle persone non presenti di poter rimanere al passo su ciò che è stato svolto, e su ciò che dovranno fare. Questo resoconto verrà comunicato tramite Telegram, così da avere informazioni istantanee senza aspettare la stesura del verbale;

- stesura del Verbale interno da parte di un membro del gruppo fornendo una breve descrizione sui punti focali dell'incontro svolto.

#### 4.1.5.2.2 Incontri esterni (con verbale)

Negli incontri esterni i partecipanti sono i membri del gruppo e il referente individuato dall'azienda in quel momento. Questi vengono richiesti dal *Responsabile di Progetto*, che valuterà come organizzare e mediare l'incontro, decidendo una corretta scansione degli eventi e della comunicazione, coinvolgendo almeno un timekeeper (colui che tiene il tempo e organizza come definire botta e risposta nella comunicazioni) e uno scriba (colui che tiene traccia di quanto è stato detto), al fine di massimizzare gestione del tempo e tracciamento degli *highlights*, momenti importanti dell'incontro.

Gli incontri avvengono in modalità virtuale, secondo lo strumento Slack, come semplice videochiamata coinvolgente alcuni membri del gruppo presenti in quel momento e il referente dell'azienda. Tramite i canali di comunicazione presenti, si dettagliano i momenti salienti dell'incontro e si tiene conto di quanto detto tramite verbale.

#### 4.1.5.2.3 Reperibilità membri

Ogni membro del nostro gruppo si impegna ad essere reperibile per riunioni sincrone durante la settimana dal lunedì al venerdì, durante il pomeriggio. Ove non fosse possibile che una o più persone siano disponibili nella data stabilita, verrà avvertito il Responsabile di Gruppo. Inoltre la reperibilità può essere estesa durante il weekend per motivi di necessità.

Durante il progetto, ogni membro è libero di svolgere i suoi compiti individuali asincroni che gli sono stati assegnati come meglio crede, rispettando le scadenze fissate e tenendo in considerazione altri impegni accademici e/o personali.

### 4.1.6 Rotazioni dei ruoli

É prevista una rotazione dei ruoli a cadenza periodica da parte del gruppo *SWEG*.

L'attribuzione dei ruoli viene svolta secondo i seguenti criteri:

- **Equità;**
- **Continuità;**
- **Assenza di conflitti.**

### 4.1.7 Gestione dei compiti e task

Il gruppo, nell'attività di progetto, è organizzato per far fronte agli impegni presenti e calendarizzare correttamente impegni ed attività, secondo la scansione definite dalle milestone, per come definito dal *Piano di Progetto v. 2.0.0* dal *Responsabile di Progetto*.

Per task di processi primari e di supporto, si utilizza Jira, in particolare dettagliando il seguente ciclo di vita delle azioni svolte:

- creazione, in cui l'attività viene definita come compito da svolgere sotto forma di ticket su Jira;
- assegnazione, in cui a discrezione della calendarizzazione operata dal *Responsabile*, vengono assegnati compiti e ruoli, come documentato anche dalle comunicazioni interne attraverso Telegram;
- completamento, l'attività viene completata da chi sta svolgendo il compito in quel momento, per poi essere ripresa, attraverso un sistema di notifica in un secondo momento;
- verifica, utilizzando un *Verificatore* come calendarizzato per effettuare un controllo della qualità delle azioni effettuate;
- accettazione, a seguito dell'azione confermata dal *Verificatore*, il *Responsabile* esegue un controllo e, a propria discrezione, comunica l'esito della verifica a chi di dovere e fa passare di stato le singole attività.

Le singole task vengono commisurate in base alla loro dimensione e alla calendarizzazione individuata come utile, sia a livello di carico che di responsabilità. In particolare, si individuano e seguono le categorizzazioni presenti nel documento, dunque processi primari, di supporto ed organizzativi, questi ultimi coadiuvati dal supporto dei Verificatori.

#### 4.1.7.1 Metodo organizzativo

Il gruppo, al fine di minimizzare ritardi e massimizzare lo svolgimento delle proprie attività, ha deciso di implementare in modo Agile<sup>6</sup> le tecniche di miglioramento continuo, implementate secondo il framework Scrum<sup>6</sup>, che permette di dividere il tempo di lavoro in intervalli piccoli, frammentati all'interno di un periodo di circa due settimane, definito come sprint. In particolare, all'interno di questo, vengono svolte un numero di attività commisurate e che devono essere completate al suo interno. Individuiamo dunque una serie di fasi:

- *Sprint Planning*, con pianificazione del periodo attuale da svolgere il primo giorno dello sprint, secondo una fase di raccolta delle idee (brainstorming), definendo quali saranno le attività che dovranno essere completate nello stesso sprint (backlog), mettendo a preventivo gli impegni e il preventivo orario/personale da impiegare;
- *Sprint Review*, riunioni di revisione a cui tutti partecipano alla definizione degli obiettivi. Alla fine, dovrebbe essere prodotto almeno un incremento, cioè un software utilizzabile. In particolare, in questa vengono definiti a consuntivo le risorse impiegate commisurate agli obiettivi, suddivisi in raggiunti e non raggiunti, capendo cosa va migliorato e cosa non è stato fatto e poter aggiungere ulteriori obiettivi per lo sprint successivo;
- *Sprint Retrospective*, concludendo definitivamente lo sprint appena svolto, valutandone il generale andamento e capendo cosa è stato fatto bene e cosa può essere migliorato. In questo modo, è meglio definire come ripianificare le attività, decidendo come iniziare/continuare/concludere attività presenti o future da realizzare.

A seguito di queste fasi, è possibile determinare artefatti, quindi oggetti di interesse da esaminare e visualizzare gli incrementi possibili, all'interno dei backlog di prodotto (*product backlog*) e dello sprint svolto nel suo complesso (*Sprint Backlog*).

#### 4.1.7.1.1 Sprint Planning

Questa fase è suddivisa nelle seguenti attività:

- **Brainstorming delle idee:** tutti i membri del gruppo espongono le loro idee su ciò che ha più rilevanza nell'immediato futuro;
- **Obiettivi e issue:** il Responsabile crea lo Sprint Backlog dove verranno definiti gli obiettivi per lo sprint. Inoltre andrà ad inserire le issue corrispondenti su GitHub;
- **Preventivi:** tutti i membri del gruppo indica il proprio preventivo orario che dedicherà allo sprint.

#### 4.1.7.1.2 Sprint Review

Questa fase è suddivisa nelle seguenti attività:

- **Consuntivo e produttività individuale:** tutti i membri del gruppo espongono il lavoro fatto, compreso le attività senza successo, durante il periodo dello sprint. In caso i membri del gruppo potranno esporre i propri dubbi in una riunione live;
- **Obiettivi raggiunti:** viene stilata una lista degli obiettivi raggiunti;
- **Obiettivi non raggiunti:** viene stilata una lista degli obiettivi non portati a termine durante lo sprint. Questa lista andrà ad aggiungersi agli obiettivi da svolgere nello sprint successivo.

## 4.2 Infrastruttura

L'infrastruttura organizzativa contiene gli strumenti utilizzati per i processi organizzativi che hanno permesso di attuare in modo pratico ed efficace il lavoro.

### 4.2.1 Strumenti

#### GitHub

GitHub è un servizio web e cloud-based che aiuta gli sviluppatori ad archiviare e gestire il loro codice e a tracciare e controllare le modifiche.

#### Discord

É stato il principale strumento utilizzato dal nostro gruppo per la comunicazione interna sincrona.

Il nostro gruppo ha adibito un server personale per tutta la durata del progetto. A sua volta, tale server è stato strutturato in 2 categorie di canali:

- *Canale Vocale:* è stato il canale principale e più utilizzato durante gli incontri interni;
- *Canale Testuale:* canale utilizzato principalmente durante gli incontri interni per la condivisione di risorse.

Nelle fasi di sviluppo del PoC<sup>G</sup> e del prodotto, tali canali sono stati specializzati in canali dedicati allo sviluppo front-end, back-end, dedicato ai documenti e per la riunione generale.

### **Telegram**

Strumento di comunicazione interna asincrona utilizzato dove abbiamo creato una chat condivisa con tutti i membri per comunicazioni di vario genere, per lo più riguardanti l'organizzazione. Al suo interno, le comunicazioni rilevanti sono state "attaccate" (pinned), per consentire al gruppo di risalire facilmente alle nuove suddivisioni dei ruoli e per consentire una più facile navigazione delle comunicazioni dati gli impegni.

### **Slack**

Canale testuale asincrono per avere una comunicazione diretta tra il nostro gruppo e l'azienda proponente.

### **Jira**

Software scelto dal nostro gruppo per pianificare, monitorare e gestire il progetto tramite la creazione di una roadmap.

### **Google Calendar**

Calendario condiviso per pianificazione di incontri esterni con l'azienda.

### **Google Drive**

Strumento utilizzato come directory condivisa per documenti in via di sviluppo e quindi non ancora approvati.

### **Google Meet**

Strumento di videochiamata utilizzato per comunicazioni esterne e per formazione.

### **Gmail**

E' stato creato l'indirizzo e-mail condiviso [swegsoftware@gmail.com](mailto:swegsoftware@gmail.com) per comunicazioni esterne.

## **4.3 Miglioramento**

Durante lo svolgimento delle attività e successiva stesura dei documenti, si è cercato di operare secondo il principio di miglioramento continuo, al fine di individuare facilmente attività, ruoli e possibili miglioramenti, cercando nuove o diverse soluzioni alle problematiche insorte. Tali valutazioni sono direttamente presenti e confrontabili in sezioni apposite all'interno del *Piano di Qualifica v. 1.0.0*, dove è possibile trovare problemi riscontrati dal gruppo e opportune soluzioni, a seconda della casistica.



## 4.4 Formazione

Al fine di operare continuativamente un miglioramento sulle attività svolte e proseguire nel mantenimento corretto delle attività in modo asincrono, occorre da parte di tutti i membri del gruppo lo studio in autonomia delle tecnologie e delle modalità operative presenti, al fine di velocizzare il processo di formazione e di conoscenza degli strumenti utilizzati. Si listano, al fine di completezza, alcune documentazioni utilizzate durante lo sviluppo, sia a livello documentale che organizzativo:

- GitHub

<https://docs.github.com/>

- Jira

<https://confluence.atlassian.com/jira>

- Git

<https://docs.github.com/en/get-started/using-git/about-git>

- Framework SCRUM

<https://scrumguides.org/scrum-guide.html>

# A Standard ISO/IEC 12207

## A.1 Processi primari

*ISO/IEC 12207* è uno standard internazionale che definisce tutti i processi necessari per lo sviluppo e la manutenzione di sistemi software. Per ogni processo esso stabilisce inoltre le attività da svolgere ed i risultati da produrre.

I processi sono divisi in tre categorie:

- processi primari;
- processi di supporto;
- processi organizzativi.

### A.1.1 Acquisizione

Questo processo descrive il prodotto/servizio che soddisfa le necessità del cliente.

È formato dalle seguenti attività:

- iniziazione;
- preparazione della richiesta di proposta;
- preparazione e aggiornamento del contratto;
- monitoraggio dei fornitori;
- accettazione e completamento.

### A.1.2 Fornitura

Questo processo ha lo scopo di fornire al cliente il prodotto/servizio che soddisfa i requisiti concordati.

È formato dalle seguenti attività:

- iniziazione;
- preparazione di una risposta;
- contratto;
- pianificazione;
- esecuzione e controllo;
- revisione e valutazione;
- consegna e completamento.

### A.1.3 Sviluppo

Questo processo ha lo scopo di sviluppare un prodotto software che sia in linea con le aspettative del cliente.

Le attività sono suddivise in:

- implementazione del processo;
- Analisi dei Requisiti di sistema;
- progettazione architetturale del sistema;
- analisi dei requisiti software;
- progettazione di dettaglio del software;
- codifica e testing del software;
- integrazione software;
- collaudo del software;
- integrazione del sistema;
- collaudo del sistema;
- installazione software.

### A.1.4 Operation

Il processo in questione viene svolto in simultanea alla fase di Manutenzione.

Esso ha lo scopo di mantenere operativo il sistema e di fornire il supporto agli utenti.

Le attività sono suddivise in:

- implementazione del processo;
- operational testing;
- system operation;
- supporto dell'utente.

### A.1.5 Manutenzione

Questo processo è svolto simultaneamente alla fase Operation e ha lo scopo di modificare il prodotto software dopo il suo rilascio per correggere eventuali difetti o migliorare alcune parti di esso.

Questo processo si conclude definitivamente con il ritiro del prodotto software.

E' suddiviso nelle seguenti attività:

- implementazione del processo;
- analisi del problema o della modifica;
- implementazione della modifica;
- revisione/accettazione della manutenzione;
- migrazione;
- ritiro del prodotto software.

## **A.2 Processi di supporto**

### **A.2.1 Documentazione**

Questo processo ha lo scopo di garantire lo sviluppo e la manutenzione delle informazioni prodotte e registrate relative al prodotto software.

### **A.2.2 Gestione della configurazione**

Il processo ha lo scopo di definire e mantenere l'integrità dei componenti della configurazione e di renderli accessibili a chi ne avesse diritto.

### **A.2.3 Gestione della qualità**

Questo processo ha lo scopo di assicurare che tutti i prodotti di fase siano conformi con gli standard ed i requisiti definiti in precedenza.

### **A.2.4 Verifica**

Questo processo ha lo scopo di confermare che i prodotti software di un'attività soddisfano i requisiti imposti dalle attività precedenti.

Il processo di Verifica dovrebbe essere integrato nei processi di Sviluppo, Fornitura e Manutenzione il prima possibile.

### **A.2.5 Validazione**

Questo processo ha lo scopo di accertare e confermare che il prodotto software rispetti i requisiti individuati.

### **A.2.6 Revisione congiunta**

Questo processo ha lo scopo di revisionare i processi eseguiti rispetto agli obiettivi definiti negli accordi. Tali revisioni devono essere svolte durante l'intero ciclo di vita, sia a livello di progetto che a livello tecnico.

Alle revisioni congiunte partecipano solo il team se viene revisionato un singolo componente. Nel caso venga effettuata una revisione sull'intero prodotto, partecipa anche il committente alla revisione.

### **A.2.7 Audit**

Questo processo ha lo scopo di determinare la conformità dei prodotti e dei processi rispetto a requisiti, pianificazione e accordi.

L'attività di auditing è svolta da personale che non ha partecipato direttamente allo sviluppo dei prodotti, dei servizi o dei sistemi oggetto delle revisioni.

### **A.2.8 Risoluzione dei problemi**

Il processo ha lo scopo di assicurare che tutti i problemi che vengono individuati, vengano risolti in modo responsabile e documentato.

## **A.3 Processi organizzativi**

### **A.3.1 Gestione organizzativa**

Il processo ha lo scopo di organizzare e controllare l'avvio e le prestazioni di un determinato processo per il raggiungimento degli obiettivi prefissati in accordo con quelli di business dell'organizzazione.

### **A.3.2 Gestione delle infrastrutture**

Questo processo ha l'obiettivo di istituire e mantenere un'infrastruttura stabile necessaria a supportare qualsiasi altro processo.

L'infrastruttura può comprendere: hardware, software, strumenti, tecniche, metodi, standard e funzionalità per lo sviluppo, operatività o manutenzione.

### **A.3.3 Miglioramento**

Questo processo ha l'obiettivo di valutare, misurare e migliorare un processo di ciclo di vita del software.

### **A.3.4 Formazione**

Il processo ha l'obiettivo di fornire e assicurarsi di mantenere personale istruito.

L'acquisizione, la fornitura, lo sviluppo e molti altri processi dipendono da personale competente ed esperto nel settore.

## B Standard ISO/IEC 9126

*ISO/IEC 9126* è uno standard internazionale che descrive un modello di qualità del software, definisce le caratteristiche che la determinano e propone metriche per la misurazione.

### B.1 Modello di qualità

Il modello di qualità definito dallo standard si divide in sei principali caratteristiche, ognuna delle quali si articola in sottocategorie, misurabili attraverso delle metriche di qualità.

Di seguito verranno descritte le sei caratteristiche sopra citate.

#### B.1.1 Funzionalità

Capacità del prodotto software di fornire funzioni in grado di soddisfare i requisiti stabiliti o implicitamente dedotti.

Si suddivide nelle seguenti sottocategorie:

- **Adeguatezza:** capacità del prodotto software di fornire un insieme di funzioni in grado di svolgere specifici task e obiettivi dell'utente;
- **Accuratezza:** capacità del prodotto software di fornire i risultati desiderati con un certo grado di precisione;
- **Interoperabilità:** capacità del prodotto software di interagire con uno o più sistemi;
- **Sicurezza:** capacità del prodotto software di proteggere informazione e dati, controllando e negando accessi o modifiche a persone o sistemi non autorizzati;
- **Aderenza alle funzionalità:** capacità del prodotto software di aderire agli standard, alle convenzioni e ai regolamenti che siano attinenti alle funzionalità offerte.

#### B.1.2 Affidabilità

Capacità del prodotto software di mantenere un dato livello di performance specificato quanto utilizzato in condizioni specifiche.

Si suddivide nelle seguenti sottocategorie:

- **Maturità:** capacità del prodotto software di evitare il verificarsi di errori e risultati non corretti durante l'esecuzione;
- **Tolleranza ai guasti:** capacità del prodotto software di mantenere il livello di prestazioni predeterminato a seguito di malfunzionamenti o usi scorretti del prodotto;
- **Recuperabilità:** capacità del prodotto software di ripristinare il livello di prestazioni e di recuperare le informazioni rilevanti in seguito ad un malfunzionamento. A seguito di un'errore, il prodotto può essere inaccessibile per un arco di tempo determinato. Questo periodo è valutato proprio dalla recuperabilità;
- **Aderenza all'affidabilità:** capacità del prodotto software di aderire a standard, regole e convenzioni inerenti all'affidabilità.

### B.1.3 Efficienza

Capacità del prodotto software di fornire prestazioni appropriate rispetto alla quantità di risorse usata durante l'esecuzione, in determinate condizioni.

Si suddivide nelle seguenti sottocategorie:

- **Comportamento rispetto al tempo:** capacità del prodotto software di fornire tempi appropriati di risposta e di elaborazione sotto specifiche condizioni;
- **Utilizzo delle risorse:** capacità del prodotto software di utilizzare un numero e tipo corretto di risorse durante l'esecuzione, in condizioni di utilizzo date;
- **Conformità:** capacità del prodotto software di aderire a standard, regole e convenzioni inerenti all'efficienza.

### B.1.4 Usabilità

Capacità del prodotto software di essere comprensibile, apprendibile, usabile e attraente nei confronti dell'utente quando viene utilizzato in condizioni specifiche.

Si suddivide nelle seguenti sottocategorie:

- **Comprensibilità:** capacità del prodotto software di essere comprensibile all'utente nei concetti base del prodotto e di come poterli utilizzare nello svolgimento di determinati compiti;
- **Apprendibilità:** capacità del prodotto software di essere comprensibile all'utente e di permettergli di impararne ad usarlo;
- **Operabilità:** capacità del prodotto software di permettere agli utenti di utilizzarlo e controllarlo;
- **Attrattività:** misura della gradevolezza del prodotto durante il suo utilizzo;
- **Aderenza all'usabilità:** capacità del prodotto software di aderire a standard, regole e convenzioni inerenti all'usabilità.

### B.1.5 Manutenibilità

Capacità del prodotto software di essere modificato, includendo correzioni e miglioramenti.

Si suddivide nelle seguenti sottocategorie:

- **Analizzabilità:** misura della difficoltà riscontrata durante la diagnosi sul prodotto ed individuare le cause di errori o malfunzionamenti;
- **Modificabilità:** facilità nell'apportare modifiche al prodotto originale (codice, progettazione, documentazione);
- **Stabilità:** capacità del prodotto software di evitare effetti indesiderati a causa di eventuali modifiche;
- **Provabilità:** capacità del prodotto software di essere verificato;
- **Aderenza alla manutenibilità:** capacità del prodotto software di aderire a standard, regole e convenzioni inerenti alla manutenibilità.

### B.1.6 Portabilità

Capacità del prodotto software di essere trasferito da un ambiente di esecuzione ad un altro.

Si suddivide nelle seguenti sottocategorie:

- **Adattabilità:** capacità del prodotto software di;
- **Installabilità:** capacità del prodotto software di essere installato in un determinato ambiente;
- **Coesistenza:** capacità del prodotto software di coesistere con altre applicazioni in ambienti comuni e condividere le risorse;
- **Sostituibilità:** capacità del prodotto software di sostituire un altro software specifico indipendente per svolgere lo stesso scopo nello stesso ambiente;
- **Aderenza alla portabilità:** capacità del prodotto software di aderire a standard e convenzioni relative alla portabilità.

## B.2 Metriche per la qualità esterna

Questa parte dello standard va a definire le metriche utili a misurare quantitativamente la qualità esterna del prodotto software.

Con qualità esterna ci si riferisce ai comportamenti del prodotto sulla base di test, dell'operatività e dall'osservazione durante la sua esecuzione.

## B.3 Metriche per la qualità interna

Questa parte dello standard va a definire le metriche utili a misurare quantitativamente la qualità interna del prodotto software. Questo tipo di metriche si applicano al prodotto non eseguibile durante la sua progettazione e codifica. Possono essere anche chiamate misure statiche.

Queste metriche permettono di individuare problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il software eseguibile.

## B.4 Metriche per la qualità in uso

Questa parte dello standard va a definire le metriche utili a misurare quantitativamente la qualità in uso del prodotto software. Queste metriche sono utili a misurare il grado con cui il prodotto software permetti agli utenti di svolgere le proprie attività con efficacia, produttività, soddisfazione e sicurezza.

La qualità in uso rappresenta il punto di vista che l'utente ha del prodotto ed è misurata in base ai risultati ottenuti dal suo utilizzo.