



B.smart - Progetto *Etherless*

Norme di Progetto

Informazioni essenziali

Nome documento	Norme di Progetto
Versione	1.0.0
Stato	Approvato
Redazione	Andrea Didonè Maria Morra
Verifica	Maurizio Andres Baggio Denis Benato
Approvazione	Matteo Lucato
Uso	Interno
Distribuzione	B.smart
Destinato a	B.smart Prof. Tullio Vardanega Prof. Riccardo Cardin
Riferimento email	b.smart.swe@gmail.com



Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
1.0.0	2020-04-12	Matteo Lucato	Responsabile	Approvazione documento
0.6.0	2020-04-12	Denis Benato	Verificatore	Verifica §A
0.5.3	2020-04-11	Maria Morra	Amministratore	Stesura §A.3
0.5.2	2020-04-10	Andrea Didonè	Amministratore	Stesura §A.2
0.5.1	2020-04-10	Maria Morra	Amministratore	Stesura §A.1
0.5.0	2020-04-09	Maurizio Andres Baggio	Verificatore	Verifica §5
0.4.2	2020-04-08	Maria Morra	Amministratore	Stesura §5.2
0.4.1	2020-04-07	Andrea Didonè	Amministratore	Stesura §5.1
0.4.0	2020-04-05	Denis Benato	Verificatore	Verifica §4
0.3.2	2020-04-04	Andrea Didonè	Amministratore	Modifica §4.2 e stesura §4.3, §4.4
0.3.1	2020-04-03	Maria Morra	Amministratore	Stesura §4.1, §4.2
0.3.0	2020-04-01	Maurizio Andres Baggio	Verificatore	Verifica §3
0.2.4	2020-03-31	Andrea Didonè	Amministratore	Stesura §3.4, §3.5
0.2.3	2020-03-30	Maria Morra	Amministratore	Stesura §3.2, §3.3
0.2.2	2020-03-30	Andrea Didonè	Amministratore	Completamento stesura §3.1
0.2.1	2020-03-29	Andrea Didonè	Amministratore	Stesura §3.1.1, §3.1.2, §3.1.3, §3.1.4
0.2.0	2020-03-28	Denis Benato	Verificatore	Verifica §2
0.1.2	2020-03-27	Maria Morra	Amministratore	Stesura §2.2



0.1.1	2020-03-26	Maria Morra	Amministratore	Stesura §2.1
0.1.0	2020-03-25	Maurizio Andres Baggio	Verificatore	Verifica §1
0.0.2	2020-03-24	Andrea Didonè	Amministratore	Stesura §1
0.0.1	2020-03-23	Andrea Didonè	Amministratore	Creazione del template in L ^A T _E X



Indice

1	Introduzione	8
1.1	Scopo del documento	8
1.2	Scopo del prodotto	8
1.3	Glossario	8
1.4	Riferimenti	8
1.4.1	Riferimenti Normativi	8
1.4.2	Riferimenti Informativi	8
2	Processi Primari	9
2.1	Fornitura	9
2.1.1	Obiettivo	9
2.1.2	Attività	9
2.1.2.1	Studio di Fattibilità	9
2.1.2.2	Piano di Progetto	9
2.1.2.3	Piano di Qualifica	10
2.1.3	Strumenti	10
2.2	Sviluppo	12
2.2.1	Obiettivo	12
2.2.2	Attività	12
2.2.2.1	Analisi dei Requisiti	12
2.2.2.2	Progettazione	12
2.2.2.3	Codifica	13
2.2.3	Strumenti	14
3	Processi di Supporto	16
3.1	Documentazione	16
3.1.1	Obiettivo	16
3.1.2	Descrizione	16
3.1.3	Ciclo di vita	16
3.1.4	Struttura dei documenti	16
3.1.4.1	Frontespizio	16
3.1.4.2	Registro delle modifiche	17
3.1.4.3	Indice	17
3.1.4.4	Contenuto principale	17
3.1.5	Template	17
3.1.6	Norme tipografiche	17
3.1.6.1	Stile di testo	17
3.1.6.2	Elenchi puntati	18
3.1.6.3	Elenchi enumerati	18
3.1.6.4	Formati comuni	18
3.1.6.5	Sigle	18
3.1.7	Elementi grafici	19
3.1.7.1	Tabelle	19
3.1.7.2	Figure	20
3.1.8	Classificazione dei documenti	20
3.1.8.1	Documenti interni	20
3.1.8.2	Documenti esterni	20
3.1.8.3	Verbali	20



3.1.8.4	Glossario	21
3.1.9	Strumenti	21
3.1.9.1	LaTeX	21
3.1.9.2	Overleaf	21
3.1.9.3	TeXstudio e Visual Studio Code	21
3.1.9.4	draw.io	21
3.2	Gestione della configurazione	22
3.2.1	Versionamento	22
3.2.2	Tecnologie	22
3.2.2.1	Git	22
3.2.2.2	GitHub	22
3.2.2.3	Tipi di file	22
3.2.2.4	Asana	22
3.3	Gestione della qualità	23
3.3.1	Obiettivo	23
3.3.2	Attività	23
3.3.2.1	Controllo qualità di processo	23
3.3.2.2	Controllo qualità di prodotto	23
3.3.3	Strumenti	23
3.4	Verifica	24
3.4.1	Obiettivo	24
3.4.2	Attività	24
3.4.2.1	Analisi	24
3.4.2.2	Test	25
3.4.3	Strumenti	26
3.4.3.1	Verifica ortografica	26
3.4.3.2	Verifica di leggibilità	26
3.5	Validazione	26
3.5.1	Obiettivo	26
3.5.2	Attività	26
4	Processi Organizzativi	27
4.1	Gestione delle comunicazioni	27
4.1.1	Comunicazioni interne	27
4.1.2	Comunicazioni esterne	27
4.2	Gestioni delle riunioni	27
4.2.1	Riunioni interne	27
4.2.2	Riunioni esterne	27
4.2.3	Verbali delle riunioni	28
4.3	Processi di pianificazione	28
4.3.1	Ruoli di progetto	28
4.3.2	Coordinamento	29
4.4	Formazione	29
4.4.1	Guide:	29
5	Metriche	30
5.1	Metriche di Processo	30
5.1.1	MPC1 Variazione di pianificazione (VP)	30
5.1.2	MPC2 Variazione di budget (VB)	30
5.1.3	MPC3 Stima del completamento (SC)	30
5.1.4	MPC4 Variazione al completamento (VC)	30



5.1.5	MPC5 Percentuale delle ore di <i>verifica_G</i> (POV)	30
5.1.6	MPC6 Numero di rischi riscontrati (NRR)	30
5.1.7	MPC7 Numero di errori non identificati dalla <i>verifica_G</i> (ENI)	31
5.2	Metriche di Prodotto	31
5.2.1	MPD1 Indice di Gulpease (IG)	31
5.2.2	MPD2 Numero di errori ortografici trovati (EOT)	31
5.2.3	MPD3 Numero di errori ortografici residui (EOR)	31
5.2.4	MPD4 Completezza funzioni sviluppate (CFS)	31
5.2.5	MPD5 Accuratezza funzioni sviluppate (AFS)	31
5.2.6	MPD6 Correttezza delle interfacce e dei protocolli sviluppati (CIP)	31
5.2.7	MPD7 Rilevamento dei difetti (RD)	32
5.2.8	MPD8 Copertura dei test (CT)	32
5.2.9	MPD9 Grado di recuperabilità (GR)	32
5.2.10	MPD10 Qualità della messaggistica (QM)	32
5.2.11	MPD11 Tolleranza agli errori commessi dagli utenti (TEU)	32
5.2.12	MPD12 Completezza delle build del test (CBT)	32
A	<i>Standard_G</i> di qualità	33
A.1	ISO/IEC 15504	33
A.2	<i>Ciclo di Deming_G</i>	34
A.3	ISO/IEC 90126	34
A.3.1	Il modello di qualità	34
A.3.2	Metriche interne ed esterne	36



Elenco delle tabelle

3.4.1	Tabella dei rischi	25
-------	------------------------------	----



Elenco delle figure

2.1.1	Immagine presa dal sito del programma: www.ganttproject.biz	10
2.1.2	Immagine dell'interfaccia Excel	11



1 Introduzione

1.1 Scopo del documento

Questo *documento_G* definisce le convenzioni che il gruppo decide di adottare sull'utilizzo degli strumenti di sviluppo, l'organizzazione e le norme sullo stile di codifica e di scrittura. Grazie a questo il gruppo avrà a disposizione le linee guida da seguire consentendogli di lavorare in modo omogeneo e facilitare il processo di *verifica_G*. Questo documento sarà soggetto a modifiche.

1.2 Scopo del prodotto

Etherless_G è una *DApp_G* accessibile attraverso una *CLI_G*, che ha lo scopo di consentire agli utenti della piattaforma la pubblicazione del proprio software in ambiente *serverless_G*, permettendo a tutti gli utilizzatori del servizio di eseguirlo pagando un corrispettivo attraverso la rete *Ethereum_G*.

1.3 Glossario

Al fine di evitare possibili ambiguità relative al linguaggio utilizzato nei documenti formali, viene fornito il *Glossario 1.0.0*. In questo documento vengono definiti e descritti tutti i termini tecnici e/o specifici che necessitano di un possibile approfondimento. Per facilitare la lettura, i termini saranno contrassegnati in corsivo e da una *G* a pedice.

1.4 Riferimenti

1.4.1 Riferimenti Normativi

- **Capitolato_G C2 - Etherless:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C2.pdf>;
- **ISO/IEC 12207:1995:**
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;
- **ISO 8601:**
<https://www.w3.org/TR/NOTE-datetime>.

1.4.2 Riferimenti Informativi

- **Documentazione AWS Lambda:**
<https://aws.amazon.com/it/lambda/>;
- **Ethereum:**
<https://ethereum.org/it/>;
- **Ruoli e i loro compiti:**
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/R0.html>;
- **L^AT_EX:**
<https://www.latex-project.org/help/documentation>.



2 Processi Primari

2.1 Fornitura

2.1.1 Obiettivo

Questa sessione ha lo scopo di identificare le richieste e i bisogni del *proponente_G*, esaminare le risorse e gli strumenti al fine di ottenere il ruolo di *fornitori_G* del prodotto, definire una pianificazione per la fornitura del prodotto finale e determinare le proprietà e le *metriche_G* del sistema.

2.1.2 Attività

2.1.2.1 Studio di Fattibilità

Uno dei ruoli principali del Responsabile di Progetto è l'organizzazione di riunioni per dare al gruppo la possibilità di comunicare e di confrontare le varie proposte di capitolato. La documentazione riguardante l'analisi e le motivazioni per cui uno è stato scelto a discapito di altri è inserita nello *Studio di Fattibilità 1.0.0* e va ad indicare per ogni *capitolato_G*:

- **Descrizione del progetto:** viene identificato il proponente e la sua richiesta;
- **Obiettivo del progetto:** vengono descritte le caratteristiche principali del progetto e ciò che si vuole ottenere;
- **Tecnologie presenti nel progetto:** vengono elencate le tecnologie coinvolte dandone una breve descrizione;
- **Aspetti positivi:** vengono valutati gli aspetti che più hanno convinto il gruppo, per interesse, conoscenza e opportunità;
- **Criticità:** sono stati valutati i fattori negativi, che hanno spinto in gruppo verso la scelta di altri progetti;
- **Conclusione:** vengono espresse le riflessioni su interessi e dubbi da parte dei componenti del gruppo, ripercorrendo le motivazioni che hanno indirizzato la nostra scelta verso la decisione di prendere o scartare il progetto.

2.1.2.2 Piano di Progetto

Il Piano di Progetto pianifica le attività e assegna le risorse disponibili per poterle svolgere. L'*efficienza_G* nell'organizzazione di questo documento ci permette di garantire l'ottenimento di un prodotto *efficace_G*. Il documento è strutturato nel modo seguente:

- **Introduzione:** viene presentato il documento descrivendone l'obiettivo e gli aspetti che verranno affrontati in esso;
- **Analisi dei rischi:** vengono identificati, classificati e descritti i rischi che potrebbero presentarsi e vengono individuate le possibili contromisure;
- **Modello di sviluppo:** Viene definito quale modello di sviluppo sarà seguito, analizzandone gli aspetti positivi e le sue *criticità_G*;
- **Pianificazione:** Viene organizzata la suddivisione delle attività sulla base delle scadenze e dei ruoli che ogni componente del gruppo andrà a ricoprire;
- **Preventivo:** viene calcolato per ognuna delle fasi che porteranno alla conclusione del progetto. Viene definito sulla base dei ruoli che saranno ricoperti e sull'ammontare delle ore svolte;



- **Consuntivo:** viene effettuato un confronto tra la spesa preventivata e quella effettivamente raggiunta. Sulla base dell'eventuale differenza verranno modificati i successivi preventivi e riorganizzati i ruoli per le successive scadenze;
- **Organigramma:** vengono definiti e assegnati i ruoli.

2.1.2.3 Piano di Qualifica

Questo documento ha come obiettivi la documentazione delle strategie di verifica e il censimento delle misurazioni delle metriche applicate dal gruppo, al fine di facilitare, garantire e tener traccia della loro qualità.

- **Qualità di processo:** vengono determinate le metriche e i parametri da mantenere per ottenere dei processi di qualità;
- **Qualità di prodotto:** vengono determinate le metriche e i parametri da mantenere per ottenere un documento e un software di qualità;
- **Appendici:** vengono elaborati i risultati di *verifica_G*, le valutazioni di miglioramento e l'esito delle revisioni.

2.1.3 Strumenti

- Gantt Project

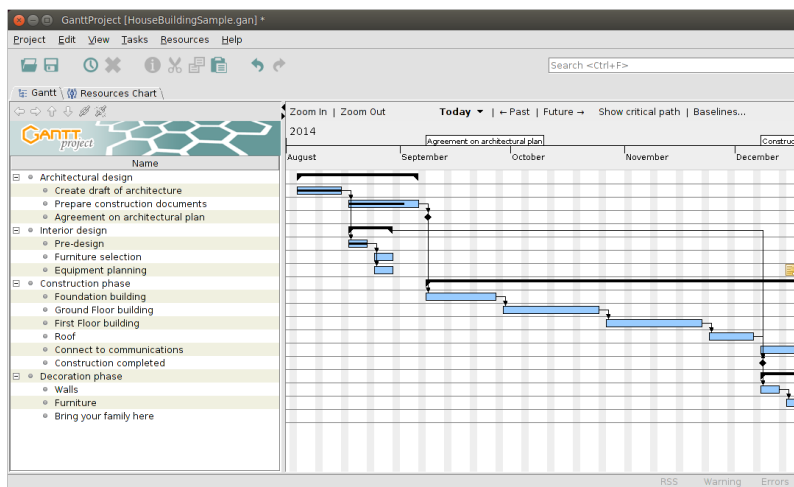


Figura 2.1.1: Immagine presa dal sito del programma: www.ganttproject.biz



- Excel

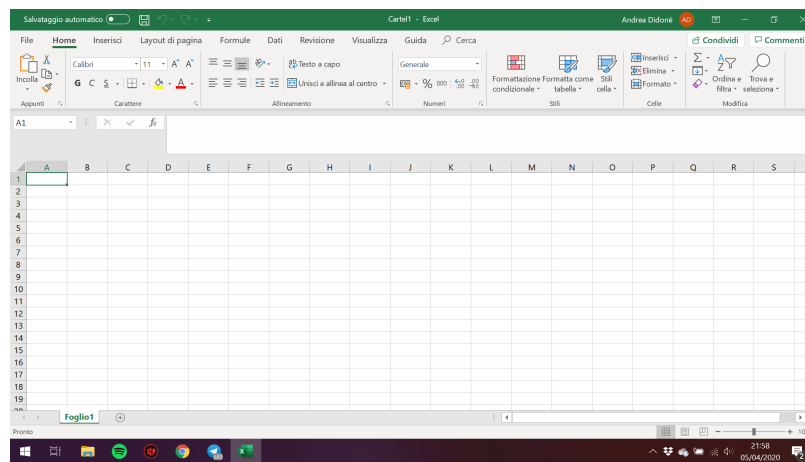


Figura 2.1.2: Immagine dell'interfaccia Excel



2.2 Sviluppo

2.2.1 Obiettivo

Il processo di sviluppo contiene tutte le attività volte a produrre il software richiesto dal *proponente_G*, Red Babel.

2.2.2 Attività

2.2.2.1 Analisi dei Requisiti

Questo documento, redatto dagli analisti, si compone di un'analisi dettagliata del prodotto che il *cliente_G* richiede. L'analisi viene effettuata sotto diversi punti di vista: vengono fissate le funzionalità e i requisiti concordati col proponente; viene effettuata una modellazione concettuale del sistema attraverso la determinazione dei casi d'uso; vengono forniti ai progettisti la classificazione e il tracciamento dei requisiti; vengono studiate operazioni di manutenzione e affinamento, per garantire un processo di sviluppo continuo del prodotto richiesto e del processo che ci permette di ottenerlo.

Casi d'uso I casi d'uso descrivono tutte le casistiche di interazione tra utenti e sistema. Vengono definiti all'interno dell'Analisi dei Requisiti e sono strutturati così:

- **Codifica:** univoca per ogni caso d'uso ed è composta da: sigla UC, un numero che identifica un'azione che l'utente può svolgere direttamente o indirettamente e un numero per ogni estensione, inclusione o sottocaso di cui un UC è composto;
- **Attori:** che si dividono in primari e secondari;
- **Descrizione:** l'attività che determina l'interazione tra le parti;
- **Scenario principale:** obiettivo e compito degli attori coinvolti;
- **Precondizione:** condizione di partenza degli attori coinvolti;
- **Postcondizione:** conseguenze e stato finale degli attori coinvolti;
- **Estensioni**(se presenti): elenco dei casi d'uso in cui è compreso quello in esame;
- **Inclusioni**(se presenti): elenco dei casi d'uso compresi in quello in esame;

2.2.2.2 Progettazione

L'obiettivo è quello di definire le caratteristiche del software richiesto, in funzione dei requisiti analizzati nel documento di Analisi dei Requisiti. La progettazione deve essere efficiente ed efficace e deve soddisfare gli *stakeholders_G*. Il risultato della progettazione è la realizzazione dell'architettura del sistema. Le parti principali sono due:

Tecnology Baseline

- Vengono determinate le tecnologie, i *framework_G*, le librerie e motivato il loro utilizzo;
- Realizzare un *PoC_G* accessibile su GitHub che dimostri come le tecnologie prese in considerazione risultino utili.



Product Baseline

- Presenta la baseline architetturale del prodotto;
- Presenta i diagrammi delle classi e di sequenza;
- Determina i design pattern conformi all'architettura.

Diagrammi I diagrammi sono utilizzati per raffigurare e chiarire le scelte progettuali effettuate. Si dividono in:

- **Diagrammi dei casi d'uso:** descrivono le interazioni tra il sistema e l'esterno;
- **Diagrammi di classe:** descrivono gli oggetti che fanno parte del sistema e come interagiscono;
- **Diagrammi di sequenza:** descrivono come più oggetti si relazionano nello scenario di un determinato comportamento;
- **Diagrammi di attività:** descrivono la logica procedurale;
- **Diagrammi di packages:** raggruppa gli elementi sulla base della visibilità rispetto ai vari packages.

2.2.2.3 Codifica

La fase di codifica riguarda il momento in cui i programmatori produrranno il codice per sviluppare il software richiesto. Costoro devono seguire le norme disposte qui di seguito, per poter garantire uniformità e per rendere il codice più leggibile, semplificando le attività di manutenzione, verifica e *validazione_G*.

Viene ritenuto necessario inoltre che la codifica rispetti gli obiettivi di qualità definiti all'interno del documento *Piano di Qualifica 1.0.0*.

Stile di codifica Questo paragrafo mette a disposizione le norme di codifica che il gruppo B.smart intende seguire. Le strutture vengono quindi così formalizzate:

- **Indentazione:** ogni blocco innestato ha un livello di indentazione diverso a seconda del livello di profondità del blocco stesso e viene utilizzato il tabulatore (TAB);
- **Parentesi:** le parentesi di apertura di un blocco sono da posizionare in linea con i costrutti, quelle di chiusura vengono poste sotto all'ultima istruzione, con livello di profondità di indentatura precedente al blocco. Nel caso in cui il blocco contiene una sola istruzione questa può essere definita inline senza l'utilizzo delle parentesi.

All'interno delle parentesi tonde non vengono utilizzati spazi, ma vengono messi prima e dopo segni tipografici di tipo matematico.

Esempi:

```
- if(val = 1) return true;

- if(val = 1) {
    return true;
}
```



- **Nomenclatura:** i nomi devono essere coerenti con lo scopo della funzione o della variabile, evitando nomi con una sola lettera. Potrebbe essere necessario utilizzare nomi composti, in questo caso fare una distinzione:

- **Nomi di funzioni e variabili:** si segue la notazione lowerCamelCase, non ci sono spazi tra le parole, l'iniziale della prima parola è minuscola mentre le altre sono maiuscole:

```
function addToVal(addend) {  
    return val+addend;  
}
```

- **Nomi di classi:** si segue la notazione UpperCamelCase, con l'iniziale di ogni parola scritta in maiuscolo:

```
class RegisteredUser{  
    ...  
}
```

- **Lingua:** nomi di variabili, classi e metodi devono essere in inglese, così come i commenti.

2.2.3 Strumenti

Di seguito gli strumenti e le tecnologie richieste e scelte per la realizzazione del progetto:

- **Servizi**

- **AWS Lambda_G:** piattaforma di calcolo senza server guidata dagli eventi fornita da Amazon come parte di Amazon Web Services;
- **Serverless Framework:** framework Web open source scritto utilizzando Node.js *serverless_G*;

- **Linguaggi e pacchetti**

- **ESLint:** strumento di analisi del codice statico per identificare i modelli problematici trovati nel codice *JavaScript_G*;
- **NodeJS:** runtime di *JavaScript_G* open source multiplatforma orientato agli eventi per l'esecuzione di codice *JavaScript_G*;
- **Solidity:** linguaggio di programmazione orientato agli oggetti per la scrittura di *smart contract_G*;
- **Typescript 3.8:** Linguaggio di programmazione che estende *JavaScript_G* aggiungendo la *tipizzazione_G* obbligatoria delle variabili;
- **Web3.js:** Libreria *JavaScript_G* che permette l'interazione tramite protocolli *HTTP_G* e *IPC_G* con un nodo *Ethereum_G* remoto;

- **Blockchain e ambienti di test**

- **Ethereum:** piattaforma decentralizzata per la creazione e pubblicazione peer-to-peer di *smart contract_G*;
- **Smart Contract:** *protocolli_G* informatici che facilitano, verificano, o fanno rispettare, la negoziazione o l'esecuzione di un contratto;
- **Ganache:** *Blockchain_G* di test utilizzata per lo sviluppo dell'applicazione. Verrà qui testata pubblicamente l'applicazione in staging;



- **Truffle:** *Framework_G* nato per facilitare lo sviluppo di *smart contracts_G* e *DApp_G*. Utilizzato inoltre come ambiente di test dell'applicativo in locale.



3 Processi di Supporto

3.1 Documentazione

3.1.1 Obiettivo

L'obiettivo di questa sezione è di delineare gli *standard_G* che il gruppo seguirà per effettuare la stesura, le modifiche e il mantenimento dei documenti durante l'intero svolgimento del progetto.

3.1.2 Descrizione

Questo capitolo contiene le norme che permetteranno di garantire uniformità, ordine e comprensione, agevolando il processo di verifica e *approvazione_G*.

3.1.3 Ciclo di vita

Il ciclo di vita di ogni documento segue le seguenti fasi:

- **Creazione:** il documento viene creato seguendo le norme previste e ordinato sulla base di un *template_G* condiviso dal gruppo. Esso è composto da: un frontespizio, un registro modifiche, un indice che guida il lettore sui contenuti del documento ed infine il contenuto stesso;
- **Stesura:** dopo che il Responsabile di Progetto ha assegnato le varie sezioni ai Redattori, questi scrivono il documento seguendo un modello incrementale;
- **Verifica:** procedimento effettuato dai Verificatori. Costoro potranno confermare la correttezza del documento oppure notificare eventuali errori o incongruenze. Nel caso in cui vengono riscontrate delle inesattezze il documento viene riassegnato;
- **Approvazione:** fa parte dei ruoli del Responsabile riconoscere l'approvazione del documento il quale viene poi rilasciato.

3.1.4 Struttura dei documenti

3.1.4.1 Frontespizio

Si tratta della pagina iniziale di ogni documento e contiene:

- **Logo:** il primo elemento visibile è il logo ufficiale completo, che si distingue dal simbolo che viene invece inserito nello *header_G* delle pagine successive;
- **Gruppo e progetto:** elementi posizionati centralmente e sotto il logo;
- **Titolo:** il nome del documento;
- **Informazioni essenziali:** una tabella contenente le seguenti informazioni:
 - **Nome documento:** nome del documento;
 - **Versione:** versione del documento;
 - **Stato:** fase del ciclo di vita a cui appartiene il documento;
 - **Redazione:** nome e cognome dei componenti del gruppo che hanno il compito di redigere il documento;
 - **Verifica:** nome e cognome dei componenti del gruppo che hanno il compito di verificare il documento;

- **Approvazione:** nome e cognome del responsabile che hanno il compito di approvare il documento;
- **Uso:** l'utilizzo può essere "interno" o "esterno";
- **Distribuzione:** il distributore del documento;
- **Destinato a:** i destinatari del documento;
- **Riferimento email:** la mail del gruppo B.smart, b.smart.swe@gmail.com.

3.1.4.2 Registro delle modifiche

La pagina successiva al frontespizio contiene il registro delle modifiche in cui vengono segnate le versioni stese del documento e le attività svolte su di esso. Il registro contiene:

- Lo storico delle versioni redatte;
- La data in cui ogni versione è stata modificata;
- Il nominativo di colui che ha modificato;
- Il ruolo svolto dalla persona che ha modificato;
- La descrizione di ciò che è stato modificato.

3.1.4.3 Indice

L'indice elenca i contenuti del documento divisi in sezioni, sottosezioni e paragrafi, dandone una visione macroscopica degli argomenti trattati e della loro gerarchia. Nel caso il documento contenga immagini e/o tabelle, saranno presenti rispettivamente degli indici di immagini e/o indici di tabelle.

3.1.4.4 Contenuto principale

La struttura destinata al contenuto del documento è comune a tutti i documenti ed è così composta:

- **Intestazione:** contiene il simbolo del gruppo sulla sinistra, mentre sulla destra è inserito il titolo del documento e la sezione numerata in cui ci si trova;
- **Corpo:** ha dei margini, in altezza e in larghezza, definiti e comuni a tutti i documenti per mantenere omogeneità;
- **Piè di pagina:** contiene il numero della pagina, posto sulla destra.

3.1.5 Template

Come anticipato, i documenti condividono la stessa struttura per garantire omogeneità durante tutto il processo di stesura. Per rendere il processo più efficiente abbiamo creato un template \LaTeX , riutilizzabile da tutti i membri del gruppo e adatto a tutti i documenti, contenente le strutture fin qui descritte.

3.1.6 Norme tipografiche

3.1.6.1 Stile di testo

- **Grassetto:** viene utilizzato sia per titoli di sezioni, sottosezioni e paragrafi, sia per elementi di un elenco puntato quando ritenuto necessario;
- **Corsivo:** viene utilizzato per le parole inserite nel glossario, per riferimenti ad altri documenti;
- **Maiuscolo:** solamente gli acronimi sono interamente in maiuscolo, i ruoli (esempio Responsabile di Progetto) e documenti (esempio Norme di Progetto).



3.1.6.2 Elenchi puntati

Un elenco di primo livello è raffigurato con il punto, uno di secondo livello con un trattino. Per l'iniziale della prima parola di un elenco è consigliabile usare il minuscolo, ma c'è la possibilità di utilizzare il maiuscolo se fosse ritenuto necessario enfatizzare un determinato concetto, come nel caso di una definizione. Ogni voce termina con un punto e virgola ";" mentre l'ultima con un punto ".".

3.1.6.3 Elenchi enumerati

Per gli elenchi in cui è ritenuto di particolare importanza l'ordine delle voci viene utilizzato un elenco enumerato e risponde alle stesse norme degli elenchi puntati per quanto riguarda l'iniziale della prima parola e la punteggiatura a termine di ogni voce.

3.1.6.4 Formati comuni

Per quanto riguarda il formato di date ed orari è stato deciso di seguire lo standard ISO 8601, quindi:

- **Data:**

YYYY-MM-DD

- **YYYY:** l'anno viene rappresentato con quattro cifre;
- **MM:** il mese viene rappresentato con due cifre;
- **DD:** il giorno viene rappresentato con due cifre;

- **Orario:**

hh:mm

- **hh:** l'ora è rappresentato da due cifre ed è un valore compreso tra 00 e 23;
- **mm:** i minuti sono rappresentati da due cifre ed è un valore compreso tra 00 e 59.

3.1.6.5 Sigle

Classificate per documenti, ruoli e concetti, le sigle previste sono le seguenti:

- **Documenti interni:**

- **SdF:** Studio di Fattibilità;
- **NdP:** Norme di Progetto;

- **Documenti esterni:**

- **AdR:** Analisi dei Requisiti;
- **PdP:** Piano di Progetto;
- **PdQ:** Piano di Qualifica;
- **MU:** manuale utente;
- **MS:** manuale sviluppatore;



- **Revisioni:**

- **RR:** revisione dei requisiti;
- **RP:** revisione di progettazione;
- **RQ:** revisione di qualifica;
- **RA:** revisione di accettazione;

- **Ruoli:**

- **Re:** Responsabile di Progetto;
- **Am:** Amministratore di Progetto;
- **An:** Analista;
- **Ve:** Verificatore;
- **Pr:** Programmatore;
- **Pt:** Progettista;

- **Concetti:**

- **PoC:** Proof of Concept;
- **PB:** Product Baseline;
- **TB:** Technology Baseline;

- **Test:**

- **TA:** test di accettazione;
- **TI:** test di integrazione;
- **TR:** test di regressione;
- **TS:** test di sistema;
- **TU:** test di unità;

- **Altro:**

- **NI:** non implementato.

3.1.7 Elementi grafici

3.1.7.1 Tabelle

Ogni tabella dev'essere centrata e richiede un identificativo e una breve didascalia sotto di essa che spieghi brevemente il contenuto. La nomenclatura per identificare univocamente ogni tabella è la seguente:

1. tabella;
2. sezione e sottosezione di appartenenza;
3. numero della tabella presente in quella determinata sezione, sottosezione.

Esempio:

Tabella 3.1.3

questa codifica rappresenta la terza tabella appartenente alla sezione 3.1.



3.1.7.2 Figure

Come le tabelle, anche le immagini devono essere centrate e dotate di codice identificativo e didascalia. Vengono valutate come figure i diagrammi di Gantt, UML e grafici. La loro nomenclatura è la seguente:

1. figura;
2. sezione e sottosezione di appartenenza;
3. numero dell'immagine presente in quella determinata sezione, sottosezione.

Esempio:

Figura 4.2.2

questa codifica rappresenta la seconda figura appartenente alla sezione 4.2.

3.1.8 Classificazione dei documenti

3.1.8.1 Documenti interni

Sono definiti documenti interni i documenti la cui lettura non è destinata a entità esterne. Fanno parte di questa classificazione lo Studio di Fattibilità e le Norme di Progetto.

3.1.8.2 Documenti esterni

Sono definiti documenti esterni i documenti la cui lettura è destinata al committente. Fanno parte di questa classificazione il Piano di Progetto, il Piano di Qualifica e Analisi dei Requisiti.

3.1.8.3 Verbalì

I Verbalì sono documenti la cui stesura è affidata ad uno dei membri scelto all'inizio dell'incontro e vengono poi approvati dal Responsabile di progetto. Sono divisi in interni, quando solo i membri del gruppo B.smart partecipano, e esterni, in caso di partecipazione di *stakeholders_G*. Ogni verbale contiene:

- **Luogo:** luogo fisico o software utilizzato per la comunicazione;
- **Data:** data dell'incontro, in conformità col formato YYYY-MM-DD;
- **Orario:** ora di inizio e fine incontro, in formato hh:mm;
- **Partecipanti:** nominativo dei componenti del gruppo e delle figure esterne ad esso, che partecipano all'incontro. Per queste ultime viene indicato tra parentesi il ruolo svolto, ad esempio "committente";
- **Argomenti affrontati:** vengono descritti i punti affrontati durante l'incontro e catalogate le decisioni prese, in modo tale da reperirle come fonti all'interno della documentazione nel momento in cui vengono citate.

I verbalì saranno salvati esprimendo la tipologia(interni o esterni) e la data in cui è stato svolto l'incontro. Esempio:

verbaleInterno2020-04-01



Per permettere di catalogare i punti argomentati durante le riunioni e rendere reperibili le loro soluzioni vengono codificati in questo modo:

- una sigla, che sarà VI in caso di verbale interno, VE se esterno, ed un trattino basso (.) a separarla del resto della codifica;
- un valore numerico che definisce a quale verbale appartiene, seguito da un punto;
- un valore numerico che contraddistingue gli argomenti appartenenti allo stesso verbale.

Per formulare un esempio: VI_2.3 consiste nel terzo argomento trattato nel secondo verbale interno.

3.1.8.4 Glossario

Si tratta di un documento destinato all'uso esterno. Contiene l'elenco delle parole che potrebbero essere sconosciute o che potrebbero portare ad ambiguità nella comprensione. Ogni parola è seguita dunque dalla sua definizione e verrà segnata in corsivo con una "G" a pedice.

3.1.9 Strumenti

3.1.9.1 \LaTeX

Lo strumento scelto per la stesura dei documenti è \LaTeX per i seguenti motivi:

- Ci permette di mantenere uniformità nel formato grazie ad un template condiviso da tutti i documenti;
- Supporta la modularità, rendendolo di facile gestione e organizzazione;
- Ci permette di lavorare contemporaneamente su capitoli differenti;
- Gestisce in modo autonomi indici e riferimenti.

3.1.9.2 Overleaf

Overleaf è un editor \LaTeX online che fornisce un vasto reparto di librerie gratuite e la capacità di un rendering real time facilitando l'intero processo di redazione del documento.

3.1.9.3 TeXstudio e Visual Studio Code

TeXstudio è l'*IDE* scelto per compilare i documenti scritti in \LaTeX e visualizzarli in formato PDF, mentre Visual Studio Code è stato scelto come editor di testo in quanto permette, attraverso estensioni, il supporto al linguaggio \LaTeX e l'integrazione al compilatore di TeXstudio.

3.1.9.4 draw.io

Per la produzione dei diagrammi *UML* si è scelto di usare draw.io, utilizzabile online al link <https://draw.io/>



3.2 Gestione della configurazione

3.2.1 Versionamento

Affinché un documento venga redatto deve passare più fasi: stesura, verifica e approvazione. Durante queste fasi si ottengono versioni diverse di un documento, le quali devono essere sempre accessibili. Le diverse versioni dei documenti sono riconoscibili tramite il seguente formalismo:

X.Y.Z

dove:

- **X**: rappresenta il numero di volte in cui il documento è stato approvato dal Responsabile di Progetto;
- **Y**: rappresenta il numero di volte in cui il documento è stato verificato dai Verificatori e torna a zero quando il documento viene approvato;
- **Z**: rappresenta il numero di volte in cui il documento è stato modificato dal redattore e torna a zero dopo ogni verifica.

3.2.2 Tecnologie

3.2.2.1 Git

Per le parti del progetto da versionare si è scelto di usare il sistema di *versionamento*_G distribuito Git usando il servizio offerto da GitHub per ospitare le varie *repository*_G. Git verrà utilizzato sia da riga di comando che tramite interfaccia grafica, come GitHub Desktop o GitKraken.

3.2.2.2 GitHub

Abbiamo deciso di utilizzare GitHub per ospitare le repository e la scelta è stata dettata da:

- una conoscenza e un utilizzo precedente a questo progetto da parte dei membri di B.smart;
- la possibilità di lavorare in modo modulare tramite la gestione di più repository per mantenere una suddivisione chiara delle diverse parti della documentazione del progetto.

3.2.2.3 Tipi di file

Per la documentazione del progetto sono stati utilizzati:

- file con estensione .tex sorgenti di L^AT_EX;
- file con estensione .pdf che sono compilati dai file sorgenti di L^AT_EX;
- diagrammi in formato XML per rappresentare i casi d'uso;
- immagini di grafici e diagrammi.

3.2.2.4 Asana

Per l'organizzazione del lavoro, distribuzione delle attività da svolgere e determinazione di scadenze abbiamo fatto appoggio ad Asana, che ci permette di effettuare un confronto continuo con gli obiettivi che ci siamo prefissati.



3.3 Gestione della qualità

3.3.1 Obiettivo

L'obiettivo è di garantire che i prodotti i processi rispettino determinati parametri di qualità e che il proponente venga soddisfatto.

3.3.2 Attività

3.3.2.1 Controllo qualità di processo

Il gruppo sottoporrà tutti i processi al ciclo di *Deming* e terrà presente lo standard S.P.I.C.E. (ISO/IEC 15504). Per ottenere i risultati prefissati il gruppo dovrà assicurarsi che:

- I processi siano oggetto di pianificazione dettagliata;
- I processi abbiano accesso alle risorse necessarie per il loro completamento;
- I processi siano sottoposti a controllo continuo e strutturato;
- I processi evolvano in risposta alle valutazioni e alle verifiche effettuate.

Le attività che quindi saranno svolte sono quelle di:

- **Pianificazione:** permette di raggiungere la fase di consegna attraverso un impiego economico e temporale predeterminati;
- **Controllo dei rischi:** attraverso un'attenta analisi si possono prevenire situazioni di rischio e, nel caso si dovessero verificare, sapere come affrontarle;
- **Verifica:** assicura che le attività utilizzate per la loro produzione siano coerenti con i requisiti stabiliti e aderiscano ai piani concordati per la loro implementazione.

3.3.2.2 Controllo qualità di prodotto

La qualità dei prodotti del progetto è gestita e mantenuta tramite i processi di supporto:

- **Verifica:** assicura che i prodotti siano coerenti con i requisiti stabiliti e aderiscano ai piani concordati per la loro implementazione;
- **Validazione:** garantisce che il prodotto sia coerente alle richieste e soddisfi il cliente.

3.3.3 Strumenti

Gli strumenti scelti per la qualità sono: forniti dallo standard ISO/IEC 15504 e ISO-12207 e le metriche.

Gli standard di qualità vengono discussi e approfonditi nell'appendice

3.4 Verifica

3.4.1 Obiettivo

La verifica viene effettuata nel momento in cui è stata registrata un'attività sufficientemente importante da doverne confermare la correttezza. Vengono quindi analizzate tutte le operazioni svolte successivamente alla precedente verifica e si controlla che non siano stati introdotti errori nel prodotto. Il processo di verifica si effettua sia su software che su documentazione e si pone l'obiettivo di:

- seguire procedure e criteri ben definiti per garantire che le richieste a cui il progetto deve rispondere siano rispettate;
- localizzare eventuali errori nel prodotto col fine di porvi rimedio;
- alzare il livello di qualità del prodotto;
- impostare il prodotto verso la validazione finale.

3.4.2 Attività

3.4.2.1 Analisi

Due sono i metodi che possono essere seguiti per effettuare l'analisi e sono:

- **Analisi statica:** valuta e applica la correttezza su documenti e codice, intesa come assenza di errori, controllando che rispetti le norme. Questo tipo di analisi non necessita che il software venga eseguito. Le modalità con cui è possibile effettuare l'analisi statica sono:
 - **Walkthrough:** si tratta di una tipologia di analisi che non fa affidamento all'esperienza e va di conseguenza a controllare la totalità del prodotto. Colui che svolge l'analisi non sa se siano effettivamente contenuti o meno dei difetti, di che tipo possano essere o dove si possano trovare. Queste caratteristiche rendono l'analisi per walkthrough una metodologia onerosa in termini di efficienza, ma sarà necessaria nelle prime fasi del progetto vista la mancanza di esperienza la quale ci permetterà più avanti di prevedere quali anomalie possono essere trovate;
 - **Inspection:** si tratta una metodologia che lavora esattamente in senso opposto a quella del walkthrough. Questa si basa sull'esperienza di errori che sono già stati commessi in fasi di verifica precedenti. Non richiede quindi che venga esaminato il prodotto nella sua totalità ma sarà un processo mirato e ben strutturato. A una maggiore esperienza corrisponde una maggiore efficienza.

Mirando ad ottenere la possibilità di effettuare un'analisi di tipo inspection sempre più efficiente, viene redatta nel Piano di Progetto una tabella contenente gli errori che vengono rilevati durante la fase di verifica. Gli errori sono codificati con un ID e dotati di descrizione, rilevazione e grado di rischio. La codifica è rappresentata da una sigla che identifica la tipologia di rischio e che può essere:

- **RT:** nel caso si tratti di rischi tecnologici;
- **RO:** nel caso si tratti di rischi organizzativi;
- **RP:** nel caso si tratti di rischi personali;
- **RG:** nel caso si tratti di rischi di gruppo;

seguiti da un valore numerico che permette di differenziare i rischi dello stesso tipo. Qui un esempio della struttura della tabella che è contenuta nel Piano di Progetto.

Tabella 3.4.1: Tabella dei rischi

ID	Nome	Descrizione	Rilevazione	Gradi di rischio
RT1	Inesperienza tecnologica	Molte delle tecnologie richieste durante lo sviluppo del progetto sono nuove per i componenti del gruppo che, potrebbero quindi incorrere in problemi a livello operativo	Sarà compito di ciascun componente del gruppo comunicare in maniera chiara e trasparente le proprie lacune tecniche, in particolar modo il responsabile dovrà rilevare quelle eventualmente non espresse	<i>Occorrenza_G</i> : Alta <i>Pericolosità</i> : Alta
RT2	Impedimenti <i>software_G</i>	Visto il periodo di sovraccarico della linea internet causato dalla quarantena indetta per il Covid-19 potrebbe risultare difficile la comunicazione tra componenti del gruppo o con il proponente	Sarà compito di chi andrà incontro a questa problematica comunicarlo agli altri membri del gruppo	<i>Occorrenza_G</i> : Media <i>Pericolosità</i> : Bassa

- **Analisi dinamica:** viene effettuata tramite l'esecuzione del software e prevede lo svolgimento di test (i quali saranno approfonditi nella sezione successiva) che ne verifichino il corretto funzionamento e che segnalino i difetti presenti.

3.4.2.2 Test

Abbiamo evidenziato nella sezione precedente come i test siano alla base dell'analisi dinamica. Il loro scopo è quello di verificare che il codice scritto sia correttamente funzionante. L'esempio di un test ben scritto è un test ripetibile. Vengono definiti in questo modo i test che, dato uno stesso ambiente e uno stesso input, produce sempre lo stesso output. Perché i test abbiano validità devono essere ben definiti i seguenti parametri:

- **Ambiente:** il sistema hardware e *software_G* dove viene effettuato il test;
- **Stato iniziale:** lo stato del prodotto sul quale il viene effettuato il test;
- **Input:** l'input inserito;
- **Output:** l'output atteso;
- **Istruzioni aggiuntive:** istruzioni che approfondiscono le modalità di esecuzione del test e l'interpretazione dell'output che viene ottenuto.

Esistono diverse tipologie di test che si differenziano per obiettivo e oggetto di verifica:

- **Test di unità:** vengono effettuati su singole unità di *software_G*. Questi test quindi ci concentrano sul funzionamento della singola unità e non sulle loro interazioni, che saranno testate successivamente. Dati una serie di input si verifica che gli output che si ottengono corrispondano a quelli attesi.

- **Test di integrazione:** superati i test di unità, ci si concentra sulle interazioni tra le parti. Si comincia da raggruppamenti di unità più piccoli che, una volta passati i test, diventano le unità che compongono raggruppamenti sempre maggiori, per eseguire ulteriori test di integrazione. L'obiettivo è di arrivare a raggruppare tutte le parti del prodotto, raggiungendo la dimensione totale del sistema.
- **Test di sistema:** una volta integrati tutti i componenti, il sistema viene testato nella sua interezza. I test di sistema permettono agli sviluppatori di verificare che le funzionalità ed i requisiti richiesti siano stati coperti, rispettando tutte le specifiche definite nell'Analisi dei Requisiti.
- **Test di regressione:** nel momento in cui vengono effettuate delle modifiche al sistema è necessario che questo venga testato nuovamente. Vengono riproposti i test sulla parte che è stata implementata per verificare la validità delle modifiche.
- **Test di accettazione:** riguarda una procedura di test simile a quella di sistema, che però viene eseguita con la partecipazione del committente. Questo test permette di verificare il prodotto e la soddisfazione del *cliente_G*. Nel momento in cui il prodotto supera questa fase il *software_G* è pronto al rilascio.

3.4.3 Strumenti

3.4.3.1 Verifica ortografica

Per controllare l'ortografia viene utilizzato il correttore ortografico integrato il T_EXstudio che sottolinea in rosso le parole con errori e in verde le parole che vengono ripetute con breve distanza.

3.4.3.2 Verifica di leggibilità

Per controllare la qualità di leggibilità dei documenti viene utilizzato:

https://farfalla-project.org/readability_static/

3.5 Validazione

3.5.1 Obiettivo

Gli obiettivi del processo di validazione sono:

- garantire che il prodotto sia conforme alle richieste del committente;
- consentire il rilascio del prodotto finale.

3.5.2 Attività

Il processo di validazione richiede che siano svolte le seguenti attività:

- identificare gli elementi che vengono sottoposti a validazione;
- pianificare le procedure di validazione;
- applicare le procedure;
- confrontare l'esito con i risultati che attesi.

4 Processi Organizzativi

4.1 Gestione delle comunicazioni

4.1.1 Comunicazioni interne

Per le comunicazioni interne al team B.smart sono utilizzati Telegram, dove abbiamo formato un gruppo, per un confronto veloce e immediato e Discord, dove abbiamo la possibilità di catalogare le conversazioni in base alla loro tematica, rendendo il tracciamento delle informazioni condivise molto più rapido. In questa fase di analisi la formazione dei canali si divide in:

- **Generale:** in cui vengono affrontate tematiche che riguardano tutto il team;
- **Documentazione:** un canale per ogni documento: Analisi dei Requisiti, Piano di Progetto, Piano di Qualifica, Norme di Progetto, Studio di Fattibilità, Glossario e verbali;
- **File:** dedicato ai contenuti condivisi, come immagini, loghi, guide e qualsiasi materiale formativo utile, sono reperibili più velocemente perché direttamente accessibile da questo canale.

4.1.2 Comunicazioni esterne

Sono considerate come comunicazioni esterne quelle effettuate con i professori Vardanega Tullio, Cardin Riccardo, e quelle effettuate con l'azienda Red Babel. Con i professori, com'è stato per una prima fase con i proponenti, le comunicazioni avvengono tramite mail. A questo proposito abbiamo creato un account mail che ci rappresenta e che utilizziamo in queste occasioni, a cui ogni membro ha accesso:

b.smart.swe@gmail.com

Dopo aver conosciuto i proponenti e referenti Milo Ertola e Alessandro Maccagnan, su loro consiglio, abbiamo adottato un canale *Slack_G*. In questo modo abbiamo potuto avere un confronto più diretto e informale che ci ha permesso di ottenere una collaborazione efficace.

4.2 Gestioni delle riunioni

Come per le comunicazioni, anche qui possiamo fare una distinzione tra interne e esterne, ovvero tra riunioni a cui hanno partecipato solo membri di B.smart e riunioni in cui hanno presenziato figure esterne.

4.2.1 Riunioni interne

Dopo una prima riunione conoscitiva per la quale è stato utilizzato Hangouts, un software sviluppato da Google che permette di inviare messaggi, effettuare chiamate vocali o videochiamate; abbiamo cominciato ad utilizzare Discord. Questa applicazione ci ha permesso non solo di effettuare riunioni che comprendevano tutto il gruppo, ma anche chiamate tra un numero ridotto di membri nel momento in cui fosse necessario un confronto che non richiedesse la partecipazione di tutto il team.

4.2.2 Riunioni esterne

Per riunioni che comprendono i proponenti utilizziamo Hangouts. Per accordarci con le parti esterne è il Responsabile di Progetto a fare da tramite e decidere la data e l'orario della riunione in corrispondenza della disponibilità di entrambe le parti.



4.2.3 Verbalì delle riunioni

I punti discussi durante lo svolgimento di una riunione, che sia interna o esterna, vengono annotati e viene redatto un verbale. Le norme che regolano la stesura dei verbali sono descritte nel paragrafo 3.1.8.1 di questo stesso documento.

4.3 Processi di pianificazione

Questa sezione presenta i sistemi utilizzati dal gruppo B.smart per le attività di pianificazione e coordinamento. Ogni membro del gruppo è tenuto fare uso di questa documentazione per tenere traccia delle proprie responsabilità rispetto al ruolo che ricopre.

4.3.1 Ruoli di progetto

Ogni membro, in ogni fase del progetto, avrà un ruolo ben definito. Questi saranno però ruotati, in modo da consentire ad ognuno di svolgere le diverse attività che il progetto richiede. I ruoli che vengono ricoperti e i loro doveri sono:

- **Responsabile di Progetto:** è colui che gestisce le risorse umane, le attività del gruppo, la pianificazione, il coordinamento e le relazioni esterne, ricoprendo il ruolo di intermediario con committente e proponente. I suoi compiti principali sono:
 - coordinamento delle attività svolte dai componenti del gruppo;
 - gestione di risorse e criticità;
 - redige l'organigramma;
 - approvazione dei documenti;
- **Amministratore:** ha una funzione di supporto, e monitora efficienza e operatività dell'ambiente di sviluppo. I compiti che deve assolvere sono:
 - gestione del versionamento e della documentazione del progetto;
 - gestione della configurazione del prodotto;
 - dirige le infrastrutture di supporto;
 - controlla le problematiche legate alla gestione dei processi;
- **Analista:** è una figura che non sarà presente in tutte le fasi della progettazione e si occupa dell'attività di analisi, documentazione e redazione. In particolare:
 - studia il dominio e i requisiti del problema;
 - redige i documenti di Studio di Fattibilità e Analisi dei Requisiti;
- **Progettista:** è responsabile delle attività tecnologiche e tecniche del progetto. Il suo ruolo è quello di:
 - sviluppare l'architettura su cui si basa il prodotto, sfruttando le tecnologie a disposizione;
 - redige la specifica tecnica del progetto;



- **Programmatore:** è responsabile delle attività di codifica miranti alla realizzazione del prodotto. Il Programmatore deve:
 - realizzare le componenti necessarie per l'esecuzione delle prove di verifica e validazione;
 - produrre codice manutenibile;
 - implementare le decisioni del Progettista;
- **Verificatore:** è responsabile delle attività di verifica e validazione. Il Verificatore si occupa di:
 - redigere la parte retrospettiva del Piano di Qualifica che illustra l'esito e la completezza delle verifiche;
 - rilevare i problemi e difetti del prodotto, segnalandoli al responsabile dell'oggetto preso in esame.

4.3.2 Coordinamento

Per la gestione del coordinamento abbiamo deciso di utilizzare Asana e la possibilità di comporre in questa applicazione web una project board che contiene la divisione dei compiti e la loro priorità. Ogni membro del gruppo ha accesso a questa project board e segna quando il compito a lui assegnato è stato completato. Questo comporta una fase di verifica dell'oggetto eseguito e una nuova assegnazione. La persona che ha il ruolo di supervisionare questo procedimento è il Responsabile di Progetto che determina i compiti, li assegna a uno o più membri del gruppo e li approva nel momento in cui superano la fase di verifica.

4.4 Formazione

La formazione è l'attività che permette ai componenti del gruppo di apprendere le competenze necessarie per lo svolgimento di compiti e una corretta suddivisione di questi. Si tratta di un'attività fondamentale perché molte delle tecnologie che vengono usate durante le varie fasi del progetto possono rivelarsi poco conosciute o completamente sconosciute da parte dei membri del team.

4.4.1 Guide:

Queste sono le tecnologie che richiedono approfondimento autoformativo e le documentazioni consultabili:

- **GitHub:** help.github.com;
- **L^AT_EX:** www.latex-project.org/ e www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf;
- **AWS:** docs.aws.amazon.com/;
- **Ethereum:** ethereum.org/;
- **Solidity:** solidity.readthedocs.io/en/develop/;
- **Node.js:** nodejs.org/it/docs/;
- **Web3:** web3js.readthedocs.io/en/v1.2.4/;
- **ESLint:** github.com/eslint/eslint.



5 Metriche

5.1 Metriche di Processo

Valori utilizzati nel calcolo di alcune metriche:

- **Valore completato** (VC): valore del lavoro completato fino al momento del calcolo;
- **Valore pianificato** (VP): valore del lavoro che è stato pianificato al momento del calcolo;
- **Costo attuale** (CA): importo speso al momento del calcolo;
- **Costo stimato** (CS): costo stimato per completare il progetto al momento del calcolo;
- **Budget** (B): budget assegnato al progetto;
- **Ore di *verifica_G*** (OV): numero delle ore utilizzate per la *verifica_G* dei processi;
- **Ore totali** (OT): numero delle ore totali di ciascun ruolo.

Le metriche elencate di seguito vengono utilizzate per valutare *efficienza_G* ed *efficacia_G* del processo.

5.1.1 MPC1 Variazione di pianificazione (VP)

Determina l'anticipo o il ritardo nello svolgimento dei processi:

$$VP = VC - VP$$

5.1.2 MPC2 Variazione di budget (VB)

Determina se si ha superato o meno il budget assegnato allo sviluppo del progetto:

$$VB = VC - CA$$

5.1.3 MPC3 Stima del completamento (SC)

Determina il costo del progetto durante il suo svolgimento:

$$SC = CA - CS$$

5.1.4 MPC4 Variazione al completamento (VC)

Proiezione sull'eccedenza o deficit di bilancio:

$$VC = B - SC$$

5.1.5 MPC5 Percentuale delle ore di *verifica_G* (POV)

Quanto hanno inciso le ore di *verifica_G* nell'ammontare totale delle ore:

$$POV = \frac{OV * 100}{OT}$$

5.1.6 MPC6 Numero di rischi riscontrati (NRR)

Rappresenta il numero di rischi che si sono effettivamente verificati durante un determinato processo.



5.1.7 MPC7 Numero di errori non identificati dalla *verifica_G* (ENI)

Rappresenta il numero di errori che non sono stati identificati durante l'attività di *verifica_G*.

5.2 Metriche di Prodotto

Le metriche elencate di seguito vengono utilizzate per valutare *efficienza_G* ed *efficacia_G* del prodotto.

5.2.1 MPD1 Indice di Gulpease (IG)

Determina l'indice di leggibilità del testo:

$$IG = 89 + \frac{300 * num.frase - 10 * num.lettere}{num.parole}$$

5.2.2 MPD2 Numero di errori ortografici trovati (EOT)

Gli errori ortografici possono essere verificati tramite Spell Right, lo strumento integrato di Visual Studio Code oppure lo strumento per il controllo della lingua presente in Overleaf e TeXstudio.

5.2.3 MPD3 Numero di errori ortografici residui (EOR)

Rappresenta il numero di errori ortografici che non potevano essere rilevati dagli strumenti automatici e non sono stati individuati dai verificatori.

5.2.4 MPD4 Completezza funzioni sviluppate (CFS)

Misura il livello di adeguatezza delle funzioni sviluppate rispetto alle esigenze specificate:

$$CFS = \frac{FT}{F}$$

FT = numero di funzioni ritenute adeguate allo svolgimento del *task_G*;

F = numero totale di funzioni sviluppate.

5.2.5 MPD5 Accuratezza funzioni sviluppate (AFS)

Misura il livello di accuratezza dei dati sviluppati:

$$AFS = \frac{FA}{F}$$

FA = numero di funzioni sviluppate con l'accuratezza richiesta;

F = numero totale di funzioni sviluppate.

5.2.6 MPD6 Correttezza delle interfacce e dei protocolli sviluppati (CIP)

Misura il livello di correttezza delle interfacce sviluppate e dei protocolli implementati rispetto alle esigenze specificate:

$$CIP = \frac{IS}{IP}$$

IS = numero di interfacce o protocolli sviluppati correttamente;

F = numero totale di interfacce e protocolli previsti nelle specifiche.

**5.2.7 MPD7 Rilevamento dei difetti (RD)**

Misura l'*efficacia_G* nel rilevare i difetti presenti durante le diverse fasi di sviluppo del prodotto:

$$RD = \frac{DR}{DP}$$

DR = numero di difetti rilevati nelle revisioni del prodotto;

DP = numero totale di difetti previsti nella fase di sviluppo.

5.2.8 MPD8 Copertura dei test (CT)

Misura la copertura dei test:

$$CT = \frac{CP}{CN}$$

CP = numero di casi di test pianificati;

CN = numero totale di casi di test necessari a garantire il livello di copertura richiesto dalla tipologia e dalla *criticità_G* del prodotto.

5.2.9 MPD9 Grado di recuperabilità (GR)

Misura la capacità del prodotto a recuperare le condizioni di errore:

$$GR = \frac{NS}{NSP}$$

NS = numero di situazioni in cui il prodotto ripristina con successo le condizioni iniziali;

NSP = numero totale di casi di situazioni in cui è previsto si debba ripristinare la situazione iniziale.

5.2.10 MPD10 Qualità della messaggistica (QM)

Misura il grado di chiarezza, completezza e correttezza dei messaggi previsti rispetto dei messaggi previsti rispetto alle diverse condizioni gestite dal prodotto:

$$QM = \frac{MC}{MP}$$

MC = numero di messaggi che risultano chiari, completi e corretti;

MP = numero totale di messaggi previsti.

5.2.11 MPD11 Tolleranza agli errori commessi dagli utenti (TEU)

Misura il livello di tolleranza del prodotto relativamente agli errori commessi dagli utenti:

$$TEU = \frac{MC}{MP}$$

5.2.12 MPD12 Completezza delle build del test (CBT)

Misura il numero di build necessarie per eseguire i test delle funzioni: condizioni gestite dal prodotto:

$$CBT = \frac{BN}{BP}$$

BN = numero di build necessarie a completare il test di funzioni;

BP = numero totale di build previste.



A *Standard_G* di qualità

Il gruppo ha preso come riferimento diversi *standard_G* al fine di perseguire qualità all'interno del progetto. Questa sezione andrà ad indicare gli *standard_G* scelti e a descrivere le sezioni più utili ai nostri obiettivi e capacità.

A.1 ISO/IEC 15504

Lo *standard_G* ISO/IEC 15504 ha lo scopo di fornire un modello per la valutazione della qualità dei processi di sviluppo *software_G*. É anche conosciuto come SPICE (Software Process Improvement and Capability Determination) e presenta due dimensioni per interpretare un processo:

- **La Process dimension** che definisce una suddivisione dei processi coerente a quella espressa dallo *standard_G* ISO/IEC 12207:
 - Cliente/*Fornitore_G*;
 - Ingegneria;
 - Supporto;
 - Gestione;
 - Organizzazione.
- **La Capability dimension:** definisce una scala di maturità di processo contenente cinque livelli:
 - **Incompleto (0):** il processo è caotico perché con risultati e performance incomplete;
 - **Performato (1):** il processo inizia ad essere eseguito mettendo a disposizione degli input ed output;
 - **Gestito (2):** le responsabilità e la gestione del processo sono definite;
 - **Stabilito (3):** il processo è pronto per diventare un processo *standard_G* ed essere rilasciato;
 - **Prevedibile(4):** il processo è in grado di essere sottoposto a metriche e valutazioni quantitative Spesso i risultati sono predicibili;
 - **Ottimizzante (5):** il processo attua miglioramenti qualitativi e quantitativi.

I livelli di *capability* sono determinati in base ad attributi definiti nel modo seguente:

- **Performance del processo** (1.1);
- **Gestione della performance** (2.1);
- **Gestione del prodotto in output** (2.2);
- **Definizione del processo** (3.1);
- **Distribuzione del processo** (3.2);
- **Misurazione del processo** (4.1);
- **Controllo del processo** (4.2);
- **Innovazione del processo** (5.1);
- **Ottimizzazione del processo** (5.2).



Ogni attributo viene misurato in modo oggettivo tramite una graduatoria contenente quattro valori, definita (N-P-L-F):

- **Not achieved** (0 - 15%);
- **Partially achieved** (>15% - 50%);
- **Largely achieved** (>50% - 85%);
- **Fully achieved** (>85% - 100%).

A.2 *Ciclo di Deming_G*

Il *ciclo di Deming_G* (o ciclo di PDCA, acronimo dall'inglese Plan-Do-Check-Act, in italiano "Pianificare - Fare - Verificare - Agire") è un metodo di gestione e miglioramento continuo dei processi di sviluppo *software_G* iterativo, organizzato in quattro fasi. Serve per promuovere una cultura della qualità che è tesa al miglioramento continuo dei processi e all'utilizzo ottimale delle *risorse_G*. Questo strumento parte dall'assunto che per il raggiungimento del massimo della qualità sia necessaria la costante interazione tra ricerca, progettazione, test, produzione e vendita. Per migliorare la qualità e soddisfare il cliente, è necessario passare attraverso tutte e quattro le fasi costantemente, tenendo come criterio principale la qualità. La sequenza logica dei quattro punti ripetuti per un miglioramento continuo è la seguente:

- **Plan (Pianificare):** stabilire gli obiettivi e i processi necessari per fornire risultati in accordo con i risultati attesi;
- **Do (Fare):** attuare il piano, eseguire il processo, creare il prodotto e raccogliere i dati da destinare alla fase di "Check" e "Act";
- **Check (Verifica):** studiare i risultati, misurati e raccolti nella fase del "Do" confrontandoli con i risultati attesi, obiettivi del "Plan", per verificarne le eventuali differenze. Le informazioni raccolte sono la base per realizzare il passo successivo : "Act";
- **Act (Agire):** Analizzare le differenze per determinarne le cause e applicare le modifiche per ottenere il miglioramento del processo o del prodotto.

A.3 ISO/IEC 90126

Lo *standard_G* ISO/IEC 90126 tratta della qualità del *software_G* ed è composto da 4 sezioni che vanno a descrivere in dettaglio il modello di qualità proposto e le metriche per interpretarlo.

A.3.1 Il modello di qualità

Il modello è composto da sei caratteristiche a loro volta suddivise in attributi, che descrivono in base a quali proprietà il *software_G* preso in esame deve essere valutato.

- **Funzionalità** è la capacità di un prodotto *software_G* di fornire funzioni che soddisfano esigenze stabilite, necessarie per operare sotto condizioni specifiche:
 - **Appropriatezza:** rappresenta la capacità del prodotto *software_G* di fornire un appropriato insieme di funzioni per gli specificati compiti ed obiettivi prefissati all'*utente_G*;
 - **Accuratezza:** la capacità del prodotto *software_G* di fornire i risultati concordati o i precisi effetti richiesti;
 - **Interoperabilità:** è la capacità del prodotto *software_G* di interagire ed operare con uno o più sistemi specificati;



- **Conformità:** la capacità del prodotto *software_G* di aderire a *standard_G*, convenzioni e regolamentazioni rilevanti al settore operativo a cui vengono applicate;
- **Sicurezza:** la capacità del prodotto *software_G* di proteggere informazioni e dati negando in ogni modo che persone o sistemi non autorizzati possano accedervi o modificarli, e che a persone o sistemi effettivamente autorizzati non sia negato l'accesso ad essi;
- **Affidabilità** è la capacità del prodotto *software_G* di mantenere uno specificato livello di prestazioni quando usato in date condizioni per un dato periodo:
 - **Maturità:** è la capacità di un prodotto *software_G* di evitare che si verificano errori, malfunzionamenti o siano prodotti risultati non corretti;
 - **Tolleranza agli errori:** è la capacità di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto;
 - **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. A seguito di un errore, il *software_G* può risultare non accessibile per un determinato periodo di tempo, questo arco di tempo è valutato proprio dalla caratteristica di recuperabilità;
 - **Aderenza:** è la capacità di aderire a *standard_G*, regole e convenzioni inerenti all'affidabilità;
- **L'efficienza_G** è la capacità di fornire appropriate prestazioni relativamente alla quantità di *risorse_G* usate:
 - **Comportamento rispetto al tempo:** è la capacità di fornire adeguati tempi di risposta, elaborazione e velocità di attraversamento, sotto condizioni determinate;
 - **Utilizzo delle risorse_G:** è la capacità di utilizzo di quantità e tipo di *risorse_G* in maniera adeguata;
 - **Conformità:** è la capacità di aderire a *standard_G* e specifiche sull'efficienza;
- **L'usabilità_G** è la capacità del prodotto *software_G* di essere capito, appreso, usato e benaccetto dall'*utente_G*, quando usato sotto condizioni specificate:
 - **Comprensibilità:** esprime la facilità di comprensione dei concetti del prodotto, mettendo in grado l'*utente_G* di comprendere se il *software_G* è appropriato;
 - **Apprendibilità:** è la capacità di ridurre l'impegno richiesto agli utenti per imparare ad usare la sua applicazione;
 - **Operabilità:** è la capacità di mettere in condizione gli utenti di farne uso per i propri scopi e controllarne l'uso;
 - **Attrattività:** è la capacità del *software_G* di essere piacevole per l'*utente_G* che ne fa uso;
 - **Conformità:** è la capacità del *software_G* di aderire a *standard_G* o convenzioni relativi all'*usabilità_G*;
- **La manutenibilità** è la capacità del *software_G* di essere modificato, includendo correzioni, miglioramenti o adattamenti:
 - **Analizzabilità:** rappresenta la facilità con la quale è possibile analizzare il codice per localizzare un errore nello stesso;
 - **Modificabilità:** la capacità del prodotto *software_G* di permettere l'implementazione di una specificata modifica (sostituzioni componenti);
 - **Stabilità:** la capacità del *software_G* di evitare effetti inaspettati derivanti da modifiche errate;



- **Testabilità:** la capacità di essere facilmente testato per validare le modifiche apportate al *software_G*;
- **La portabilità** è la capacità del *software_G* di essere trasportato da un ambiente di lavoro ad un altro:
 - **Adattabilità:** la capacità del *software_G* di essere adattato per differenti ambienti operativi senza dover applicare modifiche diverse da quelle fornite per il *software_G* considerato;
 - **Installabilità:** la capacità del *software_G* di essere installato in uno specificato ambiente;
 - **Conformità:** la capacità del prodotto *software_G* di aderire a *standard_G* e convenzioni relative alla portabilità;
 - **Sostituibilità:** è la capacità di essere utilizzato al posto di un altro *software_G* per svolgere gli stessi compiti nello stesso ambiente.

A.3.2 Metriche interne ed esterne

Sul modello sono basate metriche che permettono al gruppo di selezionare e misurare gli attributi di qualità del *software_G* desiderate. Queste sono divise tra:

- **Metriche interne:** si applicano al *software_G* non eseguibile (ad esempio il codice sorgente) durante le fasi di progettazione e codifica. Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale, poiché gli attributi interni influiscono su quelli esterni e quelli in uso. Le metriche interne permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il *software_G* eseguibile;
- **Metriche esterne:** misurano i comportamenti del *software_G* sulla base dei test, dall'operatività e dall'osservazione durante la sua esecuzione, in funzione degli obiettivi stabiliti in un contesto tecnico rilevante o di business.