



Error__418

[GitHub/Error-418-SWE](#)

error418swe@gmail.com

Norme di Progetto

Norme, processi e disciplina

Informazioni

Versione	1.30.0
Uso	Interno
Stato	Approvato
Responsabile	Zaccone Rosario
Redattori	Banzato Alessio Gardin Giovanni Nardo Silvio
Verificatore	Banzato Alessio
Destinatari	Gruppo Error__418 Vardanega Tullio Cardin Riccardo

Registro delle modifiche

Ver.	Data	PR	Titolo	Redattore	Verificatore
1.30.0	02-04-2024	414	DOC-616 Revisionare sezione Gestione della Qualità	Zaccone Rosario	Banzato Alessio
1.29.1	02-04-2024	413	DOC-620 Revisionare capitolo Processo di Validazione	Nardo Silvio	Banzato Alessio
1.29.0	02-04-2024	407	DOC-618 Revisionare sezione Processo di Integrazione	Todesco Mattia	Banzato Alessio
1.28.0	28-03-2024	391	DOC-617 Revisionare sezione Processo di implementazione	Todesco Mattia	Carraro Riccardo
1.27.0	27-03-2024	400	DOC-619 Revisionare sezione Verification process	Banzato Alessio	Gardin Giovanni
1.26.0	27-03-2024	400	DOC-619 Revisionare sezione Verification process	Banzato Alessio	Todesco Mattia
1.25.1	26-03-2024	401	DOC-647 Corretto link al PdP	Nardo Silvio	Banzato Alessio
1.25.0	25-03-2024	397	DOC-621 Revisionata sezione Transition process	Nardo Silvio	Banzato Alessio
1.24.1	21-03-2024	387	DOC-622 Aggiornare descrizione repo WMS3	Banzato Alessio	Todesco Mattia
1.24.0	13-03-2024	363	DOC-167 Redigere sezione Technical Processes/Transition Process	Oseliero Antonio	Banzato Alessio
1.23.0	03-03-2024	347	DOC-164 Redigere sezione Technical Processes/Implementation Process	Oseliero Antonio	Banzato Alessio
1.22.0	25-02-2024	329	DOC-534 Aggiunte codifica e struttura nel paragrafo dei rischi	Nardo Silvio	Carraro Riccardo
1.21.1	25-02-2024	324	DOC-521 Definire metriche di prodotto nelle Norme di Progetto	Zaccone Rosario	Carraro Riccardo
1.21.0	23-02-2024	321	DOC-526 Descrivere versionamento del software	Banzato Alessio	Carraro Riccardo
1.20.0	23-02-2024	319	DOC-162 Redigere sezione Technical Processes / Design Definition process	Zaccone Rosario	Carraro Riccardo
1.19.0	23-02-2024	318	DOC-161 Redigere sezione Technical Processes/Architecture Definition process	Banzato Alessio	Carraro Riccardo
1.18.0	21-02-2024	314	DOC-511 Aggiunto paragrafo variabili in Typst	Oseliero Antonio	Carraro Riccardo
1.17.8	21-02-2024	317	DOC-494 Aggiornare il riferimento all'Analisi dei Rischi	Gardin Giovanni	Carraro Riccardo
1.17.7	21-02-2024	306	DOC-510 Aggiornare paragrafo 4.6.3 per contenuto verbali	Oseliero Antonio	Carraro Riccardo
1.17.6	18-02-2024	298	DOC-506 Uso delle variabili per riferirsi ai professori	Gardin Giovanni	Carraro Riccardo

1.17.5	18-02-2024	303	DOC-501 Uso delle variabili per riferirsi ai documenti	Gardin Giovanni	Carraro Riccardo
1.17.4	15-02-2024	281	DOC-485: Revisionare con spell checker	Todesco Mattia	Carraro Riccardo
1.17.3	15-02-2024	251	DOC-428 Revisionare paragrafo 5.4	Zaccone Rosario	Nardo Silvio
1.17.2	13-02-2024	263	DOC-464 Aggiunte date di ultimo accesso	Gardin Giovanni	Carraro Riccardo
1.17.1	12-02-2024	257	DOC-431: Revisione paragrafo 5.7	Todesco Mattia	Carraro Riccardo
1.17.0	11-02-2024	254	DOC-430: Revisione paragrafo 5.6	Todesco Mattia	Gardin Giovanni
1.16.1	10-02-2024	235	DOC-424 Revisione capitolo 4	Todesco Mattia	Nardo Silvio
1.16.0	09-02-2024	237	DOC-426 Revisionare paragrafo 5.2	Zaccone Rosario	Gardin Giovanni
1.15.2	08-02-2024	230	DOC-425 Revisionare paragrafo 5.1	Zaccone Rosario	Gardin Giovanni
1.15.1	08-02-2024	232	DOC-417 Inserita versione Glossario	Oseliero Antonio	Gardin Giovanni
1.15.0	07-02-2024	229	DOC-423: Rimozione paragrafo 3	Todesco Mattia	Nardo Silvio
1.14.1	21-01-2024	210	DOC-356 Revisione sezione Processi di supporto	Oseliero Antonio	Zaccone Rosario
1.14.0	14-01-2024	163	DOC-289 Aggiornato il link relativo al documento Analisi dei rischi nelle norme di progetto	Nardo Silvio	Carraro Riccardo
1.13.3	14-01-2024	163	DOC-289 Aggiornato il link relativo al documento Analisi dei rischi nelle norme di progetto	Nardo Silvio	Carraro Riccardo
1.13.2	14-01-2024	183	DOC-354 Revisione sezione Introduzione NdP	Zaccone Rosario	Banzato Alessio
1.13.1	04-01-2024	147	DOC-270 Aggiunto riferimento allo stato "In verifica" nel Processo di gestione della configurazione	Gardin Giovanni	Carraro Riccardo
1.13.0	03-01-2024	136	DOC-158 Redazione sezione Business or Mission Analysis process	Gardin Giovanni	Carraro Riccardo
1.12.0	30-12-2023	127	DOC-292 Revisione del capitolo processo di gestione dei rischi	Nardo Silvio	Todesco Mattia
1.11.0	22-12-2023	119	DOC-152 Aggiunto capitolo Processo di Gestione dei Rischi	Nardo Silvio	Oseliero Antonio
1.10.0	15-12-2023	105	DOC-154 Aggiunto paragrafo Information Management	Carraro Riccardo	Todesco Mattia

1.9.1	14-12-2023	99	DOC-267 Sostituzione immagine con table nel paragrafo di configurazione	Carraro Riccardo	Gardin Giovanni
1.9.0	14-12-2023	92	DOC-141 Redazione sezione Configuration Management process	Carraro Riccardo	Todesco Mattia
1.9.0	14-12-2023	92	DOC-141 Redazione sezione Configuration Management process	Carraro Riccardo	Gardin Giovanni
1.8.0	10-12-2023	81	DOC-140 Redazione paragrafo Decision management process	Carraro Riccardo	Todesco Mattia
1.8.0	10-12-2023	81	DOC-140 Redazione paragrafo Decision management process	Carraro Riccardo	Banzato Alessio
1.7.0	10-12-2023	76	DOC-139 Redazione Project assessment and control process	Banzato Alessio	Carraro Riccardo
1.7.0	10-12-2023	76	DOC-139 Redazione Project assessment and control process	Banzato Alessio	Todesco Mattia
1.6.0	09-12-2023	75	DOC-138 Redazione Organizational/ Project Enabling processes	Carraro Riccardo	Gardin Giovanni
1.5.0	04-12-2023	65	DOC-149 Project Planning Process	Banzato Alessio	Gardin Giovanni
1.4.0	03-12-2023	60	DOC-145 Human Resources management process	Gardin Giovanni	Banzato Alessio
1.3.1	30-11-2023	55	DOC-109 Rimossa variabile legacy	Gardin Giovanni	Banzato Alessio
1.3.0	30-11-2023	49	DOC-109 Scrivere la sezione Processi di accordo/Fornitura	Banzato Alessio	Gardin Giovanni
1.2.0	30-11-2023	48	DOC-44 Processi di gestione dell'infrastruttura	Gardin Giovanni	Banzato Alessio
1.1.0	20-11-2023	23	DOC-42 Scrivere il paragrafo sulla documentazione	Oseliero Antonio	Gardin Giovanni
1.0.0	18-11-2023	7	DOC-40 Pubblicazione documento	Zaccone Rosario	Oseliero Antonio

Indice dei contenuti

1 Introduzione	1
1.1 Scopo del documento	1
1.2 Scopo del progetto	1
1.3 Glossario	1
1.4 Riferimenti	1
1.4.1 Riferimenti a documentazione interna	1
1.4.2 Riferimenti normativi	1
1.4.3 Riferimenti informativi	2
2 Processi di accordo	2
2.1 Processo di fornitura	2
2.1.1 Scopo e descrizione	2
2.1.2 Rapporti con il Proponente	2
2.1.3 Documentazione prodotta	3
2.1.4 Strumenti utilizzati	4
3 Processi di ciclo di vita	4
3.1 Processi organizzativi abilitanti	4
3.1.1 Processo di gestione dei modelli di ciclo di vita	5
3.1.2 Processi di gestione dell'infrastruttura	9
3.1.3 Processo di gestione delle Risorse Umane	11
3.1.4 Processo di Gestione della Qualità	12
4 Processi di gestione tecnica	14
4.1 Processo di pianificazione di progetto	14
4.1.1 Scopo	14
4.1.2 Risultati (<i>Piano di Progetto</i>)	14
4.1.3 Attività	14
4.2 Processo di valutazione e controllo di progetto	16
4.2.1 Scopo	16
4.2.2 Risultati	16
4.2.3 Attività	16
4.3 Processo di gestione delle Decisioni	18
4.3.1 Scopo	19
4.3.2 Risultati	19
4.3.3 Attività	19
4.4 Processo di Gestione dei Rischi	21
4.4.1 Scopo	21
4.4.2 Risultati	21
4.4.3 Struttura e codifica	21
4.4.4 Attività e compiti	22
4.5 Processo di Gestione della Configurazione	23
4.5.1 Scopo	23

4.5.2 Risultati	23
4.5.3 Attività	23
4.6 Processo di Gestione delle Informazioni	31
4.6.1 Scopo	31
4.6.2 Informazioni gestite	31
4.6.3 Documentazione	32
4.6.4 Stile e convenzioni	33
4.6.5 Distribuzione delle informazioni	36
4.7 Processo di Misurazione	36
4.7.1 Scopo	36
4.7.2 Risultati	36
4.7.3 Attività	36
4.8 Processo di Controllo della Qualità	38
4.8.1 Scopo	38
4.8.2 Risultati	38
4.8.3 Attività	39
5 Processi tecnici	41
5.1 Processo di analisi della missione	41
5.1.1 Scopo	41
5.1.2 Strategia di identificazione e analisi della missione	41
5.2 Processo di definizione di bisogni e requisiti degli stakeholder	42
5.2.1 Scopo	42
5.2.2 Stakeholder	42
5.2.3 Strategia di identificazione, analisi e trasformazione dei bisogni	42
5.3 Processo di definizione dell'architettura	44
5.3.1 Scopo	45
5.3.2 Preparazione e requisiti	45
5.3.3 Individuazione dell'architettura	45
5.4 Processo di Definizione del Design	46
5.4.1 Scopo	46
5.4.2 Risultati	46
5.4.3 Attività	46
5.5 Processo di implementazione	49
5.5.1 Scopo	49
5.5.2 Risultati	49
5.5.3 Attività	49
5.6 Processo di integrazione	51
5.6.1 Scopo	51
5.6.2 Risultati	51
5.6.3 Attività	52
5.7 Processo di verifica	54
5.7.1 Scopo	54



5.7.2 Risultati	54
5.7.3 Attività	55
5.8 Processo di Transizione	56
5.8.1 Scopo	57
5.8.2 Risultati	57
5.8.3 Attività	57
5.9 Processo di Validazione	58
5.9.1 Scopo	58
5.9.2 Risultati	58
5.9.3 Attività	58
6 Tracciamento paragrafi ISO/IEC/IEEE 12207:2017	60



Indice delle immagini

Figura 1: Workflow delle automazioni	27
--	----

Indice delle tabelle

Tabella 1: Tracciamento ISO-documenti	13
Tabella 2: Esempio tracciamento modifiche	25
Tabella 3: Variabili Typst per riferirsi a termini ricorrenti	35
Tabella 4: Tracciamento paragrafi ISO/IEC/IEEE 12207:2017	60

1 Introduzione

1.1 Scopo del documento

Questo documento contiene le regole del *way of working_G* che disciplinano le attività di ogni membro del gruppo *Error_418*. Queste regole mirano a garantire coerenza, uniformità ed efficacia nel processo collaborativo, promuovendo un ambiente di lavoro strutturato ed efficiente. L'approccio adottato per la redazione di questo documento è di natura incrementale. Ciò significa che il testo è soggetto ad aggiornamenti e revisioni continue al fine di integrare progressivamente le nuove norme, le *best practices* e i cambiamenti che emergono nel corso dello sviluppo del progetto. Questa flessibilità consente al gruppo di adattarsi prontamente alle dinamiche di lavoro e alle esigenze specifiche del contesto, garantendo un documento sempre allineato alle necessità attuali del gruppo.

1.2 Scopo del progetto

Il capitolato_G C5, denominato *WMS3: Warehouse Management 3D* e aggiudicato al gruppo, ha come obiettivo la realizzazione di un sistema di gestione di magazzino_G in tre dimensioni. L'applicazione sviluppata consentirà all'utente_G di accedere a un ambiente virtuale tridimensionale, dove potrà navigare all'interno di un magazzino_G e visualizzare gli oggetti presenti nelle scaffalature. L'utente_G avrà la possibilità di cercare specifici prodotti all'interno del magazzino_G, sfruttando la visualizzazione 3D per individuare rapidamente la posizione degli articoli desiderati, potrà modificare l'assetto del magazzino_G e inviare una notifica verso l'esterno in caso ci sia il bisogno di prelevare un articolo.

1.3 Glossario

Al fine di agevolare la comprensione del presente documento, viene fornito un glossario che espliciti il significato dei termini di dominio specifici del progetto. I termini di glossario sono evidenziati nel testo mediante l'aggiunta di una "G" a pedice degli stessi:

Termine di glossario_G

Le definizioni sono disponibili nel documento *Glossario v1.3.0*.

1.4 Riferimenti

1.4.1 Riferimenti a documentazione_G interna

- Documento *Glossario v1.3.0*:
https://github.com/Error-418-SWE_G/Documents/blob/main/3%20-%20PB/Glossario_v1.3.0.pdf (ultimo accesso 25/02/2024)
- Documento *Piano di Progetto_G v1.15.0*, Sezione 2 "Analisi dei Rischi_G":
https://github.com/Error-418-SWE_G/Documents/blob/main/3%20-%20PB/Documentazione_G%20esterna/Piano%20di%20Progetto_v1.15.0.pdf (ultimo accesso 21/02/2024)

1.4.2 Riferimenti normativi

- Capitolato_G "Warehouse Management 3D" (C5) di *Sanmarco Informatica S.p.A.*:
https://www.math.unipd.it/~tullio/IS_G-1/2023/Progetto/C5.pdf (ultimo accesso 13/02/2024)

- Regolamento di Progetto:
<https://www.math.unipd.it/~tullio/IS_G-1/2023/Dispense/PD2.pdf> (ultimo accesso 13/02/2024)
- Standard ISO_G/IEC_G/IEEE 12207:2017:
<https://www.iso_G.org/obp/ui/en/#iso_G:std:iso_G-iec_G-ieee:12207:ed-1:v1:en> (ultimo accesso 26/02/2024)
- Standard ISO_G/IEC_G/IEEE 29148:2018:
<https://ieeexplore.ieee.org/servlet/opac?punumber=8559684> (ultimo accesso 13/02/2024)
- SWEBOK Chapter 6: Software Configuration Management:
http://swbokwiki.org/Chapter_6:_Software_Configuration_Management (ultimo accesso 13/02/2024)
- Specifica Unified Modeling Language_G 2:
<https://www.omg.org/spec/UML_G/> (ultimo accesso 13/02/2024)

1.4.3 Riferimenti informativi

- Documentazione_G Git_G:
<https://git_G-scm.com/docs> (ultimo accesso 13/02/2024)
- Documentazione_G Jira_G:
[https://confluence.atlassian.com/jira_G](https://confluence.atlassian.com/jira<sub>G</sub) (ultimo accesso 13/02/2024)
- Documentazione_G Typst_G:
<https://typst_G.app/docs/> (ultimo accesso 13/02/2024)
- Documentazione_G Three.js_G:
<https://threejs.org/docs/> (ultimo accesso 13/02/2024)

2 Processi di accordo

2.1 Processo di fornitura

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.1.2

2.1.1 Scopo e descrizione

Il processo di fornitura ha come obiettivo l'offerta di un prodotto o servizio che soddisfi i requisiti concordati con Proponente e Committente_G. Tra quest'ultimi e il fornitore deve essere stabilito un accordo all'interno del quale vengono definiti i requisiti, le tempistiche e i costi da rispettare. Prima di allora, il fornitore avrà effettuato un'attenta analisi del progetto proposto e dei rischi annessi alla sua realizzazione, con relative linee guida per mitigarli.

2.1.2 Rapporti con il Proponente

Il dialogo tra il gruppo *Error_418* e il Proponente dovrà essere attivo e frequente fino al termine del progetto didattico_G, in modo che si riescano a raccogliere più feedback possibili riguardo la correttezza del lavoro svolto. Questa comunicazione avverrà in due modalità:

- 1) scritta, asincrona, utilizzata per:
 - comunicazioni di breve durata;
 - condivisione di verbali e materiali informativi;
 - coordinamento.
- 2) orale, sincrona, durante i quali si affronteranno:

- feedback sul lavoro prodotto;
- chiarimenti sul capitolato_G;
- chiarimenti riguardo casi d'uso e requisiti.

I meeting avranno cadenza variabile, e saranno fissati al termine di altri incontri oppure via e-mail. Il contenuto di ogni incontro sarà raccolto all'interno del relativo verbale_G. Ognuno di questi verbali sarà validato dal Proponente tramite l'apposizione di una firma, e sarà liberamente consultabile all'interno del repository_G GitHub_G del gruppo dedicato ai documenti (<https://github.com/Error-418-SWE/Documenti/tree/main>), al percorso NomeMilestone/Documentazione_G esterna/Verbali, dove NomeMilestone è uno tra:

- 1 - Candidatura;
- 2 - RTB_G;
- 3 - PB_G.

2.1.3 Documentazione_G prodotta

In questa sezione viene descritta la documentazione_G prodotta dal gruppo nel processo di fornitura, la quale sarà resa disponibile al Proponente, *Sanmarco Informatica*, e ai Committenti, ovvero il Professor Vardanega e il Professor Cardin.

2.1.3.1 Valutazione dei capitoli

Documento nel quale il gruppo ha analizzato le diverse proposte di progetto rese disponibili dai vari proponenti. Per ogni capitolato_G vengono presentati tre paragrafi:

- 1) **Descrizione:** vengono indicati i nominativi di Proponente e Committente_G, e viene presentato l'obiettivo del progetto;
- 2) **Dominio tecnologico:** vengono elencate le tecnologie consigliate dal Proponente del capitolato_G;
- 3) **Considerazioni:** il gruppo dà la propria valutazione sul capitolato_G.

2.1.3.2 Analisi dei rischi

Nel documento *Piano di Progetto*_G vengono presentati i rischi a cui il gruppo potrebbe essere esposto durante il periodo in cui lavora al progetto. Ogni rischio viene classificato secondo tre parametri:

- **impatto:** esprime l'effetto generato dall'evento;
- **probabilità:** esprime la probabilità del verificarsi del rischio;
- **conseguenze:** effetti collaterali a breve o medio termine che il rischio può comportare.

Ad ogni rischio sono inoltre associate delle buone pratiche da seguire per mitigarlo.

2.1.3.3 Preventivo dei costi

Nel Preventivo dei costi viene esposta una tabella che presenta una previsione riguardo il numero di ore di lavoro totali, per membro e per ruolo e viene fornito un calcolo del costo totale del progetto. Prima della tabella vengono spiegate le motivazioni che hanno portato alla suddivisione oraria individuata, effettuando una tripartizione del periodo di lavoro complessivo e analizzando ogni ruolo presente all'interno del gruppo.

2.1.3.4 Lettera di presentazione

Breve documento dove il gruppo si presenta e dichiara il suo impegno nello svolgimento del capitolato $_G$ scelto. Viene fornito un riferimento al repository $_G$ dove si potranno trovare i documenti necessari alla candidatura e vengono dichiarati il preventivo a finire e la data di consegna prevista.

2.1.3.5 Analisi dei Requisiti $_G$

In questo documento vengono raccolti tutti gli Use Case $_G$ e requisiti individuati dal gruppo con il supporto del Proponente. Ogni Use Case $_G$ e requisito $_G$ è identificato da un codice, così da essere facilmente individuabile e tracciabile. All'inizio del documento sono inoltre descritti i criteri di qualità che il gruppo ha seguito durante la redazione.

2.1.3.6 Piano di Progetto $_G$

Documento che governa la pianificazione dell'avanzamento del progetto, determinando task $_G$ e obiettivi da raggiungere e analizzando il lavoro svolto. È articolato in cinque sezioni:

- Rischi e loro mitigazione;
- Divisione temporale di sviluppo;
- Preventivo dei costi di realizzazione;
- Pianificazione del lavoro;
- Consuntivo del progetto.

2.1.3.7 Piano di Qualifica $_G$

Nel Piano di Qualifica $_G$ vengono fissati obiettivi di qualità e vengono descritti i processi necessari per conseguirli con relative procedure di controllo.

2.1.3.8 Glossario

Nel Glossario vengono elencati e definiti in modo preciso tutti i termini rilevanti utilizzati all'interno del progetto. È un documento estremamente importante per evitare situazioni di ambiguità, e garantire così una corretta comprensione della documentazione $_G$ da parte di tutte le parti coinvolte.

2.1.4 Strumenti utilizzati

In questa sezione sono indicati gli strumenti utilizzati dal gruppo nel processo di fornitura.

- **Zoom $_G$** : applicazione per videoconferenze;
- **Google Slides**: servizio cloud offerto da Google, utilizzato dal gruppo per le presentazioni del diario di bordo, ovvero l'attività in cui il gruppo aggiorna il Committente $_G$ riguardo l'andamento del lavoro;
- **Jira $_G$** : Issue Tracking System $_G$ utilizzato per la pianificazione del lavoro.

3 Processi di ciclo di vita

3.1 Processi organizzativi abilitanti

I processi organizzativi abilitanti hanno la funzione di garantire la capacità dell'organizzazione di acquisire e fornire prodotti o servizi attraverso l'avvio, il supporto e il controllo di progetti.

Questi processi forniscono l'infrastruttura e le risorse necessarie a supportare i progetti e il conseguimento degli obiettivi dell'organizzazione e degli accordi fra parti. Non sono da intendersi come un insieme esaustivo di processi aziendali atti alla gestione strategica dell'organizzazione.

I processi organizzativi abilitanti implementati dal gruppo sono i seguenti:

- 1) processo di gestione dei modelli di ciclo di vita;
- 2) processo di gestione dell'infrastruttura.

3.1.1 Processo di gestione dei modelli di ciclo di vita

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.2.1

3.1.1.1 Scopo

Lo scopo del processo di gestione del modello di ciclo di vita è definire, mantenere e garantire regole, processi, modelli e procedure di ciclo di vita.

Questo processo fornisce politiche, processi, modelli e procedure del ciclo di vita coerenti con gli obiettivi dell'organizzazione, che sono definiti, adattati, migliorati e mantenuti per supportare le esigenze individuali dei progetti all'interno del contesto dell'organizzazione, e che sono in grado di essere applicati mediante metodi e strumenti efficaci e comprovati.

Il gruppo adotta il modello PDCA (Plan-Do-Check-Act) per la gestione del ciclo di vita del software. Questo modello prevede quattro fasi:

- 1) Pianificazione (Plan): definizione degli obiettivi e dei processi necessari per raggiungerli;
- 2) Esecuzione (Do): attuazione del piano;
- 3) Verifica (Check): monitoraggio e misurazione dei processi e dei prodotti rispetto agli obiettivi e ai requisiti, e reportistica dei risultati;
- 4) Attuazione (Act): azioni per migliorare le prestazioni, se necessario, in base ai risultati della verifica.

3.1.1.2 Organizzazione del processo

3.1.1.2.1 Pianificazione

La pianificazione del processo è compito del Responsabile, il quale al termine dello sprint_G precedente, in base alle attività svolte e ai risultati ottenuti, pianifica le attività da svolgere nello sprint_G successivo.

La pianificazione è un'attività iterativa, che viene svolta all'inizio dello sprint_G. Il Responsabile, in relazione al progresso svolto, può decidere di modificare la pianificazione iniziale, aggiungendo o rimuovendo attività.

Questo processo permette di individuare le attività da svolgere, le risorse necessarie e le tempistiche di svolgimento, mediante il sistema di ticketing offerto da Jira_G.

Il risultato di questo processo è visibile all'interno del documento *Piano di Progetto_G*.

3.1.1.2.2 Esecuzione

Identifica il processo di sviluppo del prodotto, dove quanto pianificato viene concretamente svolto.

Il processo di esecuzione è composto da due attività principali:

- 1) sviluppo del prodotto;
- 2) sviluppo della documentazione_G.

Durante questo processo, ogni ruolo svolge le attività assegnate seguendo quanto stabilito nella pianificazione. Sarà compito del Responsabile verificare che le attività siano svolte correttamente e nei

tempi previsti. Ogni membro avrà la possibilità di segnalare eventuali criticità, avendo a disposizione una board apposita sulla piattaforma Miro_G.

Questo permette di avere un resoconto pronto a fine sprint_G in merito al processo di avanzamento, individuando:

- “Keep doing”: attività che hanno dato buoni risultati e che quindi vanno mantenute;
- “Things to change”: attività che hanno dato risultati non soddisfacenti e che quindi vanno modificate.

La progressione del lavoro è visibile mediante:

- retrospettiva;
- grafici di burndown_G (Jira_G);
- board di avanzamento (Miro_G).

I prodotti di questo processo, permettono dunque di procedere con la verifica e lo stabilimento delle contromisure e dei miglioramenti necessari.

3.1.1.2.3 Verifica

Al termine di ogni sprint_G, il gruppo procederà con il meeting di retrospettiva, durante il quale verranno analizzati i risultati ottenuti e le attività svolte, basandosi sui prodotti della fase di esecuzione:

1) grafici di burndown_G:

- permettono di avere una visione rapida di quanto del lavoro pianificato è stato portato a termine: si tratta di una metrica puramente quantitativa, che tiene conto del numero di ticket_G chiusi e del numero di ticket_G aperti;

2) board di avanzamento:

- “Keep doing”: attività che hanno dato buoni risultati e che quindi vanno mantenute: questo permette al gruppo di individuare il *modus operandi* più efficace ed efficiente per svolgere le attività;
- “Things to change”: attività che hanno dato risultati non soddisfacenti e che quindi vanno modificate: si tratta dell’aspetto più delicato da considerare, in quanto permette di individuare le criticità e le inefficienze del processo di sviluppo, e di conseguenza di apportare le modifiche necessarie per migliorare il processo stesso.

Questa analisi individua i miglioramenti da apportare al processo di sviluppo, stabilendo le contromisure necessarie per migliorare il processo stesso.

3.1.1.2.4 Attuazione

L’attuazione è l’ultima fase del processo di gestione del ciclo di vita, e consiste nella messa in pratica delle contromisure stabilite durante la fase di verifica.

L’obiettivo è sopperire alle mancanze e alle inefficienze del processo di sviluppo, in modo da migliorare la qualità del prodotto e la produttività del gruppo.

Diventa compito del Responsabile stabilire le attività necessarie per attuare le contromisure, e di

conseguenza di pianificare le attività da svolgere nello sprint_G successivo.

Il risultato di questo processo è visibile all'interno del documento *Piano di Progetto*_G.

3.1.1.3 Ruoli

3.1.1.3.1 Responsabile

Il Responsabile è la figura chiave che guida il progetto, assumendo il ruolo di referente principale per il gruppo e per gli stakeholder.

Le responsabilità del Responsabile includono:

- 1) coordinamento: ha il compito di supervisionare i membri del gruppo, assicurandosi che le attività vengano svolte nel rispetto delle norme identificate in questo documento;
- 2) pianificazione: stabilisce le attività da svolgere, le relative scadenze e priorità, sancendo l'inizio e la fine di ogni sprint_G;
- 3) monitoraggio e gestione dei costi: tiene sotto controllo l'andamento del progetto, stima i costi e gestisce l'analisi dei rischi_G, garantendo che il progetto rimanga entro il budget previsto;
- 4) *Norme di Progetto*_G: si occupa della stesura e dell'aggiornamento delle *Norme di Progetto*_G, che devono essere rispettate da tutti i membri del gruppo;
- 5) relazioni esterne: gestisce tutte le interazioni con il Proponente e i Committenti assicurando una comunicazione fluida ed efficace;

3.1.1.3.2 Amministratore

L'Amministratore è la figura incaricata di gestire l'ambiente di lavoro e gli strumenti utilizzati dal gruppo per tutta la durata del progetto. Ha il compito di assicurare che gli strumenti proposti ai membri del gruppo siano efficienti e favoriscano la qualità del lavoro. Monitora, assieme al Responsabile, il rispetto delle regole stabilite in questo documento e verifica che i servizi a disposizione del gruppo siano adeguati alle attività pianificate, promuovendo la produttività.

Le responsabilità dell'Amministratore includono:

- 1) redazione dei verbali: l'Amministratore è responsabile della stesura dei verbali relativi ai meeting interni ed esterni;
- 2) gestione delle risorse: si occupa dell'amministrazione delle risorse, delle infrastrutture e dei servizi necessari per l'esecuzione dei processi di supporto;
- 3) automatizzazione dei processi: determina gli strumenti necessari per automatizzare i processi, come la compilazione dei file sorgenti e il sistema di versionamento_G automatico;
- 4) risoluzione dei problemi: affronta e risolve i problemi legati alla gestione dei processi.

3.1.1.3.3 Analista

L'Analista individua i bisogni del Proponente e li trasforma in requisiti che saranno l'input delle attività successive. Il suo lavoro si svolge intensamente nel periodo di avvio del progetto, e si conclude con la stesura dell'*Analisi dei Requisiti*_G.

Il suo compito è di rilevanza in quanto un'incompleta o superficiale analisi può impattare in modo sensibile sulle attività successive, causando ritardi e costi aggiuntivi, andando a pregiudicare la qualità e la completezza del prodotto finale.

Le responsabilità dell'Analista includono:

- 1) documento di *Analisi dei Requisiti*_G: l'Analista è incaricato di redigere questo documento, che dettaglia i requisiti specifici del progetto.
- 2) interazione con i clienti: l'Analista lavora a stretto contatto con il Proponente o i Committenti per capire e studiare i loro bisogni;
- 3) classificazione dei requisiti: individua i requisiti e li classifica come funzionali e non funzionali, e come obbligatori, desiderabili o opzionali;
- 4) definizione del problema e degli obiettivi: l'Analista esamina la situazione attuale, identifica il problema e stabilisce gli obiettivi da raggiungere;
- 5) trasformazione dei bisogni in requisiti: durante la stesura dell'*Analisi dei Requisiti*_G, l'Analista converte i bisogni dei clienti in requisiti specifici per la soluzione.

3.1.1.3.4 Progettista

Il Progettista ha il compito di sviluppare una soluzione che soddisfi le esigenze identificate dall'Analista, rispettando i vincoli individuati. Il Progettista trasforma i requisiti in un'architettura che modella il problema.

Le responsabilità del Progettista includono:

- 1) sviluppo di un'architettura robusta e resistente ai malfunzionamenti;
- 2) realizzazione di soluzioni affidabili, efficienti, sostenibili e conformi ai requisiti;
- 3) decomposizione del sistema in componenti e organizzazione delle loro interazioni, ruoli e responsabilità;
- 4) garanzia di un basso grado di accoppiamento nell'architettura.

3.1.1.3.5 Programmatore

Il Programmatore è la figura più numerosa all'intero del gruppo.

Le sue responsabilità includono:

- 1) scrittura del codice sorgente, perseguendo manutenibilità e conformità a quanto stabilito dall'architettura definita dalla progettazione;
- 2) creazione di test specifici per la verifica e la validazione del codice.

3.1.1.3.6 Verificatore

Il Verificatore controlla il lavoro svolto dagli altri membri del gruppo, assicurandosi che le *Norme di Progetto*_G e le aspettative siano rispettate.

Le responsabilità del Verificatore includono:

- 1) verifica della qualità e della conformità della documentazione_G prodotta;
- 2) approvazione della documentazione_G sottoposta a verifica.

3.1.2 Processi di gestione dell'infrastruttura

3.1.2.1 Scopo

Lo scopo del processo di gestione dell'infrastruttura è fornire l'infrastruttura e i servizi a supporto dell'organizzazione per il conseguimento degli obiettivi di progetto nel corso dell'intero ciclo di vita.

Questo processo definisce, fornisce e regola i servizi, gli strumenti e le tecnologie di comunicazione e condivisione delle informazioni a supporto degli scopi dell'organizzazione.

3.1.2.2 Requisiti

L'infrastruttura è costituita dai servizi, dagli strumenti e dalle tecnologie di comunicazione e condivisione delle informazioni adottate a supporto degli scopi dell'organizzazione. L'infrastruttura risponde alle necessità di comunicazione interna ed esterna.

I requisiti dei processi di gestione dell'infrastruttura sono:

- 1) semplicità di adozione di strumenti e servizi;
- 2) accesso rapido alle informazioni, anche in mobilità;
- 3) non ridondanza tra strumenti e servizi adottati.

3.1.2.3 Infrastruttura di comunicazione

I principi della comunicazione che ispirano i processi di gestione dell'infrastruttura sono:

- 1) Comunicazione aperta: le comunicazioni avvengono in modalità pubblica, ovvero tutti i membri possono partecipare (compatibilmente con i loro impegni di progetto e sempre nel rispetto delle rispettive responsabilità). I membri del gruppo hanno accesso e possono liberamente consultare i messaggi, le eventuali registrazioni e i verbali;
- 2) Comunicazione onesta: quanto comunicato rappresenta sempre il reale stato del progetto. Nessun membro, in nessun caso, deve nascondere le criticità incontrate;
- 3) Comunicazione proattiva: comunicare con cognizione di causa, offrendo spunti concreti di discussione. Ogni comunicazione deve poter essere processata dagli interessati nel minor tempo possibile;
- 4) Comunicazione frequente: la frequenza della comunicazione permette di prendere decisioni in modo tempestivo e informato.

3.1.2.3.1 Comunicazione interna

Discord_G rappresenta il canale primario di comunicazione interna. È una piattaforma di comunicazione che fornisce:

- 1) un servizio di messaggistica istantanea che permette la suddivisione delle conversazioni in canali tematici;
- 2) un servizio di videochiamate usato per le comunicazioni sincrone interne in modalità remota.

Le comunicazioni testuali tramite messaggio istantaneo sono organizzate per argomento. Discord_G adotta il termine “canale” per designare tale suddivisione tematica. I canali attualmente in uso sono:

- canale generale (**#random**): usato per le comunicazioni informali;
- canale meeting (**#meeting**): usato per l’organizzazione dei meeting interni e la condivisione degli ordini del giorno;
- canale di riferimento (**#riferimenti**): usato come bacheca per raccogliere ed organizzare, in un unico luogo, le risorse a supporto degli scopi dell’organizzazione;
- altri canali tematici: le comunicazioni relative ad uno specifico prodotto dei processi dell’organizzazione avvengono in un canale dedicato.

La suddivisione delle comunicazioni interne in canali ha lo scopo di ridurre le distrazioni, facilitare l’accesso alle informazioni e semplificare la comunicazione interna.

Le comunicazioni sincrone in videochiamata avvengono nei cosiddetti “canali vocali”. Vengono forniti quattro canali vocali generici ad accesso libero. I membri dell’organizzazione hanno la facoltà di incontrarsi in qualsiasi momento in videochiamate interne. I canali vocali non sono organizzati tematicamente perché offrono la persistenza.

Miro_G è un servizio di collaborazione per team basato su *whiteboard*. Offre la possibilità di creare board multimediali e permette la collaborazione asincrona. È utilizzato per:

- 1) raccogliere i feedback interni da discutere durante i meeting di retrospettiva;
- 2) supportare gli incontri interni di *brainstorming*;
- 3) supportare i meeting con gli interlocutori esterni.

Oltre a Discord_G e Miro_G, l’organizzazione comunica anche tramite **Jira_G** (ITS_G) e **GitHub_G** (VCS). L’uso di questi strumenti è discusso in dettaglio nelle sezioni apposite.

3.1.2.3.2 Comunicazione esterna

Le modalità e la frequenza delle comunicazioni esterne sono da stabilirsi con i diretti interessati, secondo necessità e disponibilità degli interlocutori. Le comunicazioni esterne avvengono su due canali primari: Gmail e Zoom_G.

Gmail è il servizio di posta elettronica di Google. L’indirizzo di posta elettronica dell’organizzazione è:

error418swe@gmail.com

Viene utilizzato per tutte le comunicazioni da e verso gli interlocutori esterni. Tutti i membri dell’organizzazione possono accedere in qualsiasi momento alla casella di posta elettronica. Inoltre, tutte le conversazioni vengono inoltrate automaticamente agli indirizzi e-mail istituzionali di ciascun membro. L’indirizzo è reso noto nel frontespizio di ogni documento prodotto dall’organizzazione.

Zoom_G è un servizio di teleconferenza. A meno di accordi specifici tra le parti, l'organizzazione utilizza Zoom_G per effettuare videochiamate con gli interlocutori esterni.

3.1.2.4 Mantenimento dell'infrastruttura

Sono compiti dell'amministratore il mantenimento dell'infrastruttura, l'aggiornamento delle norme e dei processi e l'identificazione di nuovi servizi a supporto delle attività dell'organizzazione.

L'organizzazione adotta nuovi servizi, strumenti e tecnologie di comunicazione avendo cura di non introdurre ridondanza. L'organizzazione si dota di un insieme di strumenti e servizi sufficienti a coprire tutti i requisiti di comunicazione.

3.1.3 Processo di gestione delle Risorse Umane

La natura didattica del progetto riduce le prerogative del processo di gestione delle Risorse umane. Per questa ragione, l'organizzazione dichiara la:

conformance to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.2.4.

3.1.3.1 Scopo

Questo processo fornisce all'organizzazione le risorse umane necessarie all'esecuzione dei processi di ciclo di vita al fine di realizzare gli obiettivi di progetto, dell'organizzazione e degli stakeholder.

3.1.3.2 Attività e compiti

3.1.3.2.1 Identificare le competenze dei membri

L'organizzazione sottopone, ad ogni nuovo membro, un form conoscitivo atto a identificare le competenze pregresse. Il form è realizzato con **Google Forms** e include domande su:

- 1) strumenti di collaborazione;
- 2) linguaggi di programmazione;
- 3) tecnologie;
- 4) strumenti di automazione;
- 5) strumenti di controllo di versione_G.

Le risposte sono non vincolanti e non influiscono in alcun modo sulla rotazione dei ruoli, sui compiti assegnati o sull'organizzazione interna.

3.1.3.2.2 Identificare le competenze richieste

Le competenze richieste sono identificate tramite:

- 1) Analisi dei Capitolati;
- 2) Studio del dominio di progetto;
- 3) Incontri con i Proponenti e successivi colloqui.

3.1.3.2.3 Sviluppare le competenze

Lo sviluppo di nuove competenze riguarda i membri, e non i ruoli. Per questa ragione, i processi di sviluppo di competenze sono universali e condivisi.

L'organizzazione si adopera per sviluppare le competenze dei membri mediante:

- 1) attività di *peer-tutoring* in concomitanza delle rotazioni di ruolo;
- 2) pubblicazione interna di tutorial tecnici scritti (eventualmente accompagnati da brevi video, se utili a migliorare la comprensione degli argomenti trattati);

- 3) attività di *tutoring* interno su richiesta, sincrono, in base alla necessità;
- 4) attività di *mentoring* esterno su richiesta, in base alla necessità e alla disponibilità dell'interlocutore esterno;
- 5) condivisione delle best practice in sessione collettiva.

Le sessioni di tutoring sono “a sportello” ed è responsabilità dei singoli membri richiederne l'attivazione. Il responsabile, identificati i temi di maggior interesse, può espressamente richiedere che un ruolo copra le esigenze di tutoring interno tramite le modalità sopra indicate.

3.1.3.2.4 Acquisire e fornire competenze

I membri dell'organizzazione sono prestabiliti. Qualora le competenze interne all'organizzazione siano deficitarie, è richiesta l'attivazione delle attività descritte in Sezione 3.1.3.2.3.

Non sono previste variazioni della composizione dell'organizzazione, se non in via straordinaria e comunque discussa preventivamente con il Committente_G.

3.1.4 Processo di Gestione della Qualità

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.2.5

3.1.4.1 Scopo

Il processo di Gestione della Qualità ha lo scopo di garantire che i prodotti e i servizi del progetto soddisfino gli obiettivi di qualità del gruppo e i bisogni del Proponente.

3.1.4.2 Risultati

Come risultato dell'efficace attuazione del processo di Gestione della Qualità:

- vengono definiti e implementati gli obiettivi, le politiche e le procedure di Gestione della Qualità del gruppo;
- vengono stabiliti i criteri e i metodi di valutazione della qualità;
- vengono fornite risorse e informazioni al progetto per supportare l'operazione e il monitoraggio delle attività di garanzia della qualità del progetto;
- vengono raccolti e analizzati i risultati delle valutazioni della garanzia della qualità;
- le politiche e le procedure di Gestione della Qualità vengono migliorate basandosi sui risultati del progetto e dell'organizzazione.

3.1.4.3 Attività

3.1.4.3.1 Pianificazione

Il gruppo *Error_418* pone i seguenti principi di qualità:

- 1) il contenuto del branch_G principale delle repository_G (*src* e, di conseguenza, *main* per Documenti, e *dev* per WMS3) deve rispettare i criteri di qualità definiti nel *Piano di Qualifica_G* e nella Sezione 4.5. Inoltre:
 - tutti gli elementi di configurazione devono essere gestiti tramite repository_G su GitHub_G (Sezione 4.5.3.1.3.1), con il sistema di *feature_G branching*;
 - ogni modifica che vuole essere introdotta in un prodotto dovrà essere prima approvata dal Verificatore tramite il meccanismo di Pull Request_G (Sezione 4.5.3.3).
- 2) il contenuto della documentazione_G deve seguire i principi di redazione descritti nella Sezione 4.6.4;
- 3) la documentazione_G prodotta deve seguire degli standard interni e/o esterni:

- standard interni:
 - vengono definiti nel presente documento nella Sezione 4.6.3.
- standard esterni:
 - il gruppo adotta i seguenti standard ISO_G:

Standard ISO _G	Documento
12207:2017	<i>Norme di Progetto_G, Analisi dei Requisiti_G</i>
9126 1:2001	<i>Piano di Qualifica_G</i>

Tabella 1: Tracciamento ISO_G-documenti

In particolare, per ogni processo interno al presente documento viene specificato il grado di conformità allo standard ISO_G/IEC_G/IEEE 12207:2017 all'interno di una sezione dedicata (Sezione 6).

- 4) il codice prodotto deve seguire degli standard interni, definiti nella Sezione 5.5;
- 5) il codice prodotto dal gruppo deve rispettare i parametri di qualità attraverso il monitoraggio delle metriche identificate nel *Piano di Qualifica_G*
- 6) il prodotto software deve soddisfare i bisogni e le richieste del Proponente, identificate nel documento di *Analisi dei Requisiti_G* e catalogate in: obbligatorie, desiderabili e opzionali.

3.1.4.3.2 Valutazione

La valutazione della qualità dei prodotti di progetto, dell'andamento del gruppo e dell'attuazione dei principi di qualità elencati in precedenza viene effettuata nei seguenti modi:

- review_G delle Pull Request_G da parte del Verificatore:
 - verifica dell'aderenza agli standard interni ed esterni;
 - verifica della leggibilità, chiarezza e comprensibilità della documentazione_G;
 - verifica dell'assenza di errori ortografici_G o sintattici nei documenti;
 - verifica dell'assenza di errori nel codice (valutazione statica) e nell'esecuzione degli elementi software.
- testing_G del codice, come descritto nel processo di integrazione (Sezione 5.6) e nel processo di verifica (Sezione 5.7);
- revisioni complessive degli elementi di configurazione in prossimità della fine di ogni periodo di lavoro definito nella Sezione 4.1.3.2.1;
- utilizzo di metriche di qualità (Sezione 4.7) definite nel *Piano di Qualifica_G v1.5.0* e di un cruscotto di qualità_G (Sezione 4.8.3.1.3) per l'individuazione tempestiva di anomalie o situazioni critiche;
- comunicazione costante con il Proponente per raccogliere feedback e valutare l'adesione del lavoro alle sue richieste (Sezione 2.1.2).

3.1.4.3.3 Attuazione di azioni correttive

L'attività di attuazione di azioni correttive consiste nei seguenti task_G:

- 1) individuazione delle parti degli elementi soggetti a configurazione da correggere;
- 2) segnalazione immediata delle parti da correggere al Responsabile, tramite canale Discord_G pubblicamente consultabile;
- 3) individuazione delle azioni correttive da compiere;
- 4) apertura dei relativi task_G di revisione_G su Jira_G;

5) assegnazione e svolgimento dei task_G con successiva verifica in sede di Pull Request_G.

4 Processi di gestione tecnica

4.1 Processo di pianificazione di progetto

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.1

4.1.1 Scopo

Il processo di pianificazione di progetto ha come scopo la produzione e coordinazione di un piano efficace ed applicabile per garantire una corretta gestione del lavoro. La pianificazione viene inserita in un documento denominato *Piano di Progetto_G* (Sezione 4.1.2).

4.1.2 Risultati (*Piano di Progetto_G*)

Il *Piano di Progetto_G* è il documento principale che viene redatto durante questo processo, e ha il compito di governare la pianificazione e l'avanzamento del progetto. In questo documento vengono determinati i task_G e gli obiettivi da raggiungere, fornendo anche un'analisi critica del lavoro svolto fino a quel momento accompagnata dai grafici di Gantt_G e di burndown_G. Sono presenti cinque sezioni, di cui le prime quattro rientrano nel processo di pianificazione:

- 1) Rischi e loro mitigazione;
- 2) Divisione temporale di sviluppo;
- 3) Preventivo dei costi di realizzazione;
- 4) Pianificazione del lavoro;
- 5) Consuntivo del progetto.

La redazione del documento va di pari passo con l'avanzamento del progetto, in modo tale da essere sempre aggiornato alla situazione corrente del lavoro.

4.1.3 Attività

Nel processo di pianificazione sono presenti due attività principali:

- 1) definizione del progetto;
- 2) pianificazione del progetto e della gestione tecnica.

4.1.3.1 Definizione del progetto

In questa attività il gruppo deve definire tutto ciò che caratterizza il progetto, ovvero i suoi obiettivi e vincoli, sia di carattere funzionale che tecnologico. Durante la lavorazione del progetto verranno prodotti diversi output, che possono essere suddivisi nelle due macro categorie di: documentazione_G, codice. Entrambi questi prodotti dovranno essere realizzati rispettando determinate regole e processi, ed è quindi necessario che il gruppo definisca in questa attività uno o più cicli di vita da seguire.

4.1.3.2 Pianificazione del progetto e della gestione tecnica

È l'attività principale del processo, nella quale viene definita nel concreto la pianificazione.

4.1.3.2.1 Suddivisione temporale

Il gruppo ha individuato tre periodi di lavoro principali:

- raccolta e analisi dei requisiti_G: vengono delineati i requisiti che il prodotto finale dovrà rispettare tramite un continuo rapporto con il Proponente;
- sviluppo della Requirements and Technology Baseline_G (RTB_G): si studiano le tecnologie da utilizzare e si applicano le conoscenze acquisite per realizzare un PoC_G (Proof of Concept_G), ovvero un prodotto software che permetta di dimostrare la padronanza delle tecnologie selezionate ai fini dello sviluppo del progetto;
- periodo di sviluppo del Minimum Viable Product_G (MVP_G): viene progettato e implementato un prodotto software che rispetti almeno i requisiti minimi di accettazione, e che offra tutte le funzionalità richieste.

Ognuno di questi viene suddiviso a sua volta in periodi della durata di una settimana denominati *sprint*_G. Al termine di ogni *sprint*_G viene effettuato un incontro interno di retrospettiva, nel quale si analizza criticamente la settimana appena conclusa, mostrandone aspetti positivi, aspetti da migliorare e fissando obiettivi che verranno poi riportati nell'Issue Tracking System_G sotto forma di task_G. Questi andranno a comporre il *backlog*_G dello *sprint*_G successivo, e il loro progressivo completamento andrà a produrre un *burndown*_G-chart, utilizzato dal gruppo come strumento che rappresenti in modo oggettivo l'andamento del lavoro.

4.1.3.2.2 Definizione di ruoli, responsabilità e costi

Al fine di migliorare l'assegnazione del lavoro vengono definiti sei ruoli, ognuno dei quali con precise responsabilità da rispettare. Ogni membro del gruppo dovrà assumere ognuno di questi ruoli all'interno del periodo di lavoro al progetto. L'assegnazione dei ruoli avviene con frequenza bisettimanale. Di seguito viene riportata la descrizione di ogni ruolo con i relativi compiti:

- 1) **Responsabile**: è presente durante l'intero progetto, in particolare si occupa di:
 - coordinare il gruppo;
 - verificare che il lavoro proceda secondo le tempistiche e i costi stabiliti;
 - rappresentare il gruppo nei rapporti con il committente_G;
 - gestire la pianificazione di ogni *sprint*_G.
- 2) **Amministratore**: ruolo presente durante tutto il progetto. Ha il compito di:
 - predisporre e controllare il corretto utilizzo delle procedure e degli strumenti adottati;
 - implementare e mantenere gli automatismi in modo da migliorare l'efficienza del gruppo.
- 3) **Analista**: è presente principalmente nei primi due periodi del progetto. Si occupa di redigere il documento *Analisi dei Requisiti*_G, nel quale:
 - definisce i casi d'uso;
 - raccoglie e classifica i requisiti.
- 4) **Progettista**: ruolo presente principalmente negli ultimi due periodi, nei quali:
 - effettua uno studio delle tecnologie proposte, mirato alla definizione dello stack tecnologico da usare;
 - delinea l'architettura del prodotto;
 - definisce le linee guida implementative valutando le scelte più efficienti e sostenibili.
- 5) **Programmatore**: è attivo negli ultimi due periodi del progetto, nei quali:
 - si occupa della codifica del PoC_G, senza partire da una progettazione ben definita visto l'obiettivo del Proof of Concept_G;

- traduce in codice eseguibile l'architettura del prodotto finale definita dal progettista durante il periodo di sviluppo del MVP_G.

6) **Verificatore**: è presente durante l'intero progetto, e si occupa di controllare che il lavoro prodotto dal gruppo rispetti gli standard qualitativi adottati.

Ad ogni ruolo è inoltre associato un costo orario, sulla base del quale il gruppo calcola il preventivo totale del progetto e quello di ogni *sprint*_G seguito dal relativo consuntivo. Il costo orario viene calcolato in base alla sua importanza all'interno del progetto, misurata in termini di competenze e disponibilità della risorsa.

4.2 Processo di valutazione e controllo di progetto

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.2

4.2.1 Scopo

Il processo di valutazione e controllo ha lo scopo di determinare lo stato del progetto tramite la verifica dell'allineamento del lavoro con i piani definiti. Tale allineamento viene dimostrato osservando i seguenti parametri:

- tempistiche;
- costi;
- obiettivi raggiunti.

Questo viene fatto all'interno di revisioni periodiche chiamate retrospettive, e viene messo per iscritto all'interno dei verbali di tali riunioni e nei paragrafi di consuntivo del *Piano di Progetto*_G.

4.2.2 Risultati

Come risultato dell'implementazione del processo di valutazione e controllo si elencano i seguenti:

- valutazione dell'adeguatezza di ruoli, responsabilità e risorse;
- esecuzione delle azioni correttive scaturite dalle revisioni di progresso;
- viene avviata, se necessaria, la ripianificazione del progetto;
- viene autorizzata l'azione del progetto di avanzare (o meno) da un traguardo o evento programmato al successivo;
- vengono registrati gli obiettivi raggiunti.

I risultati sono espressi e analizzati nei paragrafi di consuntivo del *Piano di Progetto*_G.

4.2.3 Attività

4.2.3.1 Piano di valutazione e controllo del progetto

La valutazione e il controllo del progetto avvengono a cadenza settimanale, in corrispondenza della riunione di retrospettiva. Durante questa riunione si valuta ciò che è stato svolto durante il periodo di lavoro (*sprint*_G) che si sta concludendo, se ne identificano i problemi e si decidono soluzioni di controllo che vadano a risolvere o arginare i problemi individuati.

Oltre a queste revisioni periodiche interne, sono presenti attività nelle quali il gruppo espone la propria situazione al Proponente, tramite riunione su Zoom_G, e Committente_G, tramite le attività denominate "diario di bordo".

Inoltre, è necessario che il gruppo svolga delle revisioni tecniche con il Committente_G per avere una valutazione esterna. Queste due revisioni sono chiamate Requirements and Technology Baseline_G (o RTB_G) e Product Baseline_G (o PB_G).

4.2.3.2 Valutazione

Durante l'attività di valutazione il gruppo deve analizzare la situazione del progetto, e per fare ciò deve adottare degli strumenti che rappresentino tale situazione nel modo più oggettivo possibile. Il momento in cui si effettua questo compito è la retrospettiva settimanale. Questa si svolge tramite meeting interno su Discord_G con il supporto di una board Miro_G. Il meeting è suddiviso concettualmente in tre parti, rappresentate all'interno della board da tre riquadri:

- Keep doing: raccoglie tutti gli aspetti positivi (e di conseguenza da mantenere) dello sprint_G;
- Things to change: raccoglie tutte le problematiche incontrate durante lo sprint_G;
- To do e Improvements: raccoglie tutte le attività da svolgere nel prossimo sprint_G, alcune delle quali direttamente collegate agli elementi appartenenti a "Things to change".

Tramite Miro_G il gruppo riesce ad avere una panoramica della situazione del lavoro, che viene poi completata da Jira_G, l'Issue Tracking System_G adottato. Al suo interno il gruppo ha definito le due milestone_G esterne (RTB_G e PB_G), nelle quali ha creato delle Epic_G che rappresentano ciò che è necessario produrre per quella milestone_G, e che raccolgono tutti i task_G necessari alla produzione di tali prodotti (documenti o software).

Milestone_G e Epic_G vengono accompagnate all'interno di Jira_G da una barra di completamento che rappresenta in verde il lavoro svolto, in blu il lavoro in fase di svolgimento e lascia vuota la parte dedicata alle attività definite ma non ancora in svolgimento. Queste barre contribuiscono a fornire una rappresentazione oggettiva della situazione del progetto.

Oltre a queste, Jira_G offre la funzionalità di visualizzazione di un burndown_G-chart, ovvero un grafico che rappresenta l'andamento del lavoro all'interno di uno sprint_G in due possibili modalità:

- quantità di story_G point completati;
- quantità di issue_G completate.

Entrambe le rappresentazioni pongono nell'asse x del grafico il tempo, indicato in giorni. Questi grafici contengono inoltre una retta rappresentante l'andamento ideale del lavoro, grazie alla quale risulta più semplice verificare l'efficienza del gruppo.

Al termine dell'incontro di retrospettiva viene redatto il paragrafo di consuntivo dello sprint_G nel *Piano di Progetto*_G, nel quale, oltre a fare un resoconto dello sprint_G, si analizza il suo l'aspetto economico: in base ai ruoli impegnati e al monte ore produttivo svolto, si calcola il costo effettivo del periodo concluso, aggiornando conseguentemente il costo preventivato e il documento *Piano di Progetto*_G. Questo indicatore contribuisce ad avere un resoconto completo del progetto, e permette al gruppo di comprendere meglio come sta lavorando e se sta gestendo correttamente le risorse a sua disposizione.

4.2.3.3 Controllo

Nell'attività di controllo figurano i seguenti task_G:

- azioni correttive;

- ripianificazione;
- azioni di cambiamento dovute a richieste del Committente_G e/o Proponente;
- autorizzazione ad avanzare alla successiva milestone_G.

4.2.3.3.1 Azioni correttive

Nell'attività di controllo si intraprendono azioni correttive nei confronti dei problemi individuati.

Questi problemi possono essere di duplice natura:

- mancato raggiungimento degli obiettivi prefissati;
- miglioramenti e accortezze da adottare.

4.2.3.3.1.1 Mancato raggiungimento degli obiettivi prefissati

È necessario che alla chiusura dello sprint_G le attività ancora in fase di svolgimento vengano riportate nello sprint_G successivo, insieme a tutte quelle attività pianificate ma non ancora iniziate che sono considerate importanti. Tutte le attività che non vengono considerate importanti, ad esempio attività di cui si è rivalutato il grado di priorità, vengono riportate nel backlog_G. Una situazione di mancato raggiungimento degli obiettivi può essere sinonimo anche da una pianificazione errata e troppo ottimista, ed è quindi necessario che essa sia rivista e migliorata.

4.2.3.3.1.2 Miglioramenti e accortezze da adottare

Le soluzioni correttive vengono decise dal gruppo tramite la visualizzazione e l'analisi della board Miro_G durante la retrospettiva. Nella board infatti, come esposto nella Sezione 4.2.3.2, alcuni dei task_G raccolti rispondono direttamente ai problemi individuati nella parte di Things to change.

4.2.3.3.2 Ripianificazione

La ripianificazione ha atto quando gli obiettivi cambiano nel corso dello sprint_G o alcune ipotesi fatte in fase di pianificazione si rivelano sbagliate. La ripianificazione viene gestita tramite Jira_G, che consente di aggiornare i task_G attivi, permettendo anche la comunicazione tempestiva dei cambiamenti al gruppo.

4.2.3.3.3 Azioni di cambiamento dovute a richieste del Committente_G e/o Proponente

Le azioni di cambiamento dovute a richieste del Committente_G e/o Proponente sono recepite attraverso i canali di comunicazione con quest'ultimi (Zoom_G, mail) e vengono registrate nei rispettivi verbali. A queste azioni viene attribuita un'alta priorità per garantire massima soddisfazione nel cliente finale.

4.2.3.3.4 Autorizzazione ad avanzare alla successiva milestone_G

L'autorizzazione ad avanzare alla successiva milestone_G di progetto viene concessa dal Committente_G e/o Proponente in seguito ai colloqui pianificati su Zoom_G con quest'ultimi. Il gruppo si riserva di procedere verso la milestone_G successiva solo una volta ricevuta l'approvazione richiesta, in modo da non portare avanti difetti e problematiche che potrebbero risultare insidiosi da correggere una volta entrati in un periodo avanzato del progetto.

4.3 Processo di gestione delle Decisioni

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.3

4.3.1 Scopo

Lo scopo del processo di Gestione delle Decisioni è fornire un quadro strutturato per identificare, categorizzare e valutare le decisioni che si intendono intraprendere, selezionando la migliore in termini di benefici attesi, costi e rischi associati.

4.3.2 Risultati

Come risultato dell'efficace attuazione del processo di Gestione delle Decisioni:

- vengono identificate le decisioni che richiedono un'analisi alternativa;
- vengono identificati e valutati i percorsi di azione alternativi;
- viene selezionato un percorso di azione preferito;
- vengono identificate la risoluzione, la giustificazione della decisione e le ipotesi che portano ad una sua necessità.

4.3.3 Attività

4.3.3.1 Presentazione delle decisioni

Ogni decisione viene presentata identificandone:

- tipologia;
- obiettivo;
- soluzione proposta;
- vantaggi;
- svantaggi;
- impatto in termini di tempi e costi.

Il processo di decision making viene prevalentemente svolto nel meeting di retrospettiva, in modo da non contrastare la pianificazione dello sprint_G in corso ed evitare un eccessivo numero di meeting interni che potrebbero comportare difficoltà organizzative e un rallentamento dell'avanzamento. Alcune decisioni potrebbero richiedere il coinvolgimento di soggetti esterni, come Proponente e Committente_G, soprattutto nei casi in cui sia richiesta una figura con maggiore esperienza nel campo di riferimento. Solo decisioni critiche riguardo cambiamenti sostanziali o nuove direzioni di lavoro possono far scaturire meeting interni mirati.

Il resoconto di quanto deciso sarà visibile all'interno del verbale_G redatto a fine meeting.

4.3.3.1.1 Strategie di decision-making utilizzate

- 1) **Strategia collaborativa:** prevede che la decisione venga presentata e votata in modo collaborativo, coinvolgendo tutti i membri del gruppo e mediante una votazione con un sistema maggioritario, in modo che il risultato rappresenti la volontà della maggioranza.

Strategia utilizzata per le decisioni:

- organizzative;
- dei requisiti;
- qualitative.

- 2) **Strategia expertise decision-making:** prevede che la decisione venga presentata e analizzata mediante la consultazione di figure esterne più esperte, individuabili nel Proponente o nel Committente_G, la cui esperienza risulta determinante.

Questa strategia viene utilizzata per le decisioni:

- implementative;
- tecnologiche;
- architetturali.

4.3.3.2 Analisi delle decisioni

Le decisioni possono riguardare diversi aspetti del capitolato_G, e la loro categorizzazione è utile per individuare la migliore strategia di gestione e i ruoli coinvolti.

Le decisioni vengono dunque così classificate:

- 1) **Decisioni organizzative:** sono relative al modo di lavorare, cioè a come vengono gestiti i processi di avanzamento del progetto. Esempi notevoli sono le decisioni focalizzate alla coordinazione del gruppo o alla scelta degli strumenti da utilizzare per la gestione del progetto (ad esempio: scelta dell'ITS_G, delle piattaforme per la comunicazione e per la collaborazione):
 - documento soggetto a modifiche: *Norme di Progetto*_G
 - ruoli responsabili dell'aggiornamento: Responsabile, Amministratori;
 - strategia di decision-making: collaborativa.
- 2) **Decisioni tecnologiche:** sono relative allo stack tecnologico da adottare durante lo sviluppo del progetto:
 - documento soggetto a modifiche: *Norme di Progetto*_G
 - ruoli responsabili dell'aggiornamento: Responsabile, Progettisti;
 - strategia di decision-making: expertise decision-making.
- 3) **Decisioni sui requisiti:** sono relative ai requisiti del prodotto software. Possono riguardare aspetti funzionali e non funzionali:
 - documento soggetto a modifiche: *Analisi dei Requisiti*_G
 - ruoli responsabili dell'aggiornamento: Analisti;
 - strategia di decision-making: collaborativa.
- 4) **Decisioni di implementazione:** sono decisioni relative alla stesura del codice:
 - documento soggetto a modifiche: *Norme di Progetto*_G, documenti tecnici (diagramma delle classi);
 - ruoli responsabili dell'aggiornamento: Responsabile, Progettisti, Programmatori;
 - strategia di decision-making: expertise decision-making.
- 5) **Decisioni architetturali:** sono decisioni relative ai pattern e alle architetture riguardanti il software:
 - documento soggetto a modifiche: *Norme di Progetto*_G, documenti tecnici (diagramma delle classi, diagramma dei casi d'uso);
 - ruoli responsabili dell'aggiornamento: Responsabile, Progettisti;
 - strategia di decision-making: expertise decision-making.
- 6) **Decisioni sulla qualità:** sono decisioni relative ai controlli di qualità:
 - documento soggetto a modifiche: *Piano di Qualifica*_G
 - ruoli responsabili dell'aggiornamento: Responsabile, Verificatori;
 - strategia di decision-making: collaborativa.

4.4 Processo di Gestione dei Rischi

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.4

4.4.1 Scopo

Lo scopo del processo di Gestione dei Rischi è identificare, analizzare, trattare e monitorare costantemente i rischi, così da poterli affrontare sistematicamente durante l'intero ciclo di vita del progetto. I rischi e le relative opzioni di trattamento vengono codificate univocamente per agevolare il tracciamento e l'organizzazione strutturale.

4.4.2 Risultati

Come risultato dell'implementazione del processo di Gestione dei Rischi:

- vengono identificati e analizzati i rischi;
- vengono identificate e priorizzate le opzioni di trattamento del rischio;
- viene selezionato ed implementato un trattamento appropriato;
- i rischi vengono valutati per verificare cambiamenti di stato e progressi nel trattamento.

I risultati sono raccolti nel documento *Analisi dei Rischi_G*.

4.4.3 Struttura e codifica

4.4.3.1 Struttura gerarchica

I rischi vengono suddivisi nelle categorie principali descritte dal "Risk Breakdown Structure" PM-BOK:

- 1) rischi tecnici;
- 2) rischi esterni;
- 3) rischi organizzativi;
- 4) rischi di project management.

4.4.3.2 Codifica

I rischi vengono codificati tramite la sigla "RIS" separata mediante un carattere "-" da due valori "x" e "y", divisi a loro volta da un carattere ":".

- il valore di "x" identifica la categoria PMBOK corrispondente;
- il valore "y" identifica il numero del rischio.

Per esempio, il primo rischio della categoria "rischi esterni" avrà codice:

RIS-2.1

Le opzioni di trattamento vengono codificate tramite la sigla "OT" separata mediante un carattere "-" da tre valori "x", "y" e "z", divisi a loro volta da un carattere ":".

- il valore di "x" identifica la categoria PMBOK corrispondente al rischio di riferimento;
- il valore "y" identifica il numero del rischio di riferimento;
- il valore "z" identifica il numero dell'opzione di trattamento.

Per esempio, la prima opzione di trattamento del rischio RIS-2.1 avrà codice:

OT-2.1.1

4.4.4 Attività e compiti

4.4.4.1 Pianificazione della Gestione dei Rischi

La strategia di Gestione dei Rischi per il progetto è basata su un approccio proattivo per identificare e mitigare i rischi in tutte le fasi del suo ciclo di vita.

Ad ogni rischio individuato viene associato un profilo basato sia sulla probabilità di occorrenza che sull'impatto che essi hanno sullo stato di avanzamento dei lavori e sul progetto stesso. L'impatto può essere "lieve", "medio" o "grave" in base alla sua entità. La probabilità di occorrenza viene identificata tramite un valore intero da 1 a 5, dove 1 esprime una probabilità molto bassa, mentre 5 esprime una frequenza attesa sostenuta.

Il gruppo definisce approcci di trattamento appropriati, compresi piani di mitigazione specifici.

4.4.4.2 Gestione del profilo di rischio

1) **Definizione e registrazione delle soglie e delle condizioni di rischio:**

le soglie di rischio sono stabilite sulla base della probabilità di occorrenza e dell'impatto. I rischi con un impatto negativo elevato sono trattati in modo più rigoroso rispetto a quelli con un impatto inferiore;

2) **Creazione e mantenimento di un profilo di rischio:**

il profilo di rischio contiene tutte le informazioni necessarie per la Gestione dei Rischi, come il loro stato, le soglie, le probabilità, le azioni richieste in caso di occorrenza e le conseguenze previste. Viene aggiornato in modo tempestivo in risposta ai cambiamenti nelle condizioni del progetto;

3) **Fornitura del profilo di rischio rilevante agli stakeholder in base alle loro esigenze:**

il profilo di rischio viene all'occorrenza discusso nei meeting interni e/o esterni e tutte le parti interessate allo stato attuale dei rischi e delle azioni di trattamento possono consultare il documento *Analisi dei Rischi_G v2.0.1*.

4.4.4.3 Analisi dei rischi

Questa attività consiste nei seguenti compiti:

1) **identificazione del rischio ed associazione ad uno specifico profilo, secondo quanto descritto nella pianificazione della Gestione dei Rischi:**

l'identificazione avviene durante tutte le fasi di sviluppo. Inoltre, i rischi emergono dall'analisi delle misurazioni di qualità dei processi e del sistema software in evoluzione;

2) **valutazione di ciascun rischio rispetto alle sue soglie di accettazione:**

ogni rischio viene valutato rispetto alle soglie stabilite, il superamento delle quali attiva delle modalità di trattamento specifiche.

4.4.4.4 Trattamento dei rischi

Questa attività consiste nei seguenti compiti:

1) **definizione e registrazione delle strategie e delle misure di trattamento consigliate per ciascun rischio che superi la soglia di tolleranza.**

Queste possono includere:

- l'eliminazione del rischio;
- la riduzione della sua probabilità o gravità;

- l'accettazione del rischio.

Vengono anche registrate informazioni sull'efficacia delle alternative di trattamento;

2) **nel caso di accettazione di un rischio al di sopra della soglia, attribuzione di priorità elevata e monitoraggio costante:**

pratica necessaria per valutare la necessità di futuri interventi di trattamento del rischio o eventuali modifiche della sua priorità;

3) **coordinamento dell'azione di gestione dopo la selezione di una strategia di trattamento del rischio:**

dopo la selezione di un trattamento del rischio, vengono coordinate azioni di gestione per implementare le decisioni prese. Il processo di valutazione e controllo del progetto può essere applicato.

4.4.4.5 Monitoraggio dei rischi

Questa attività consiste nei seguenti compiti:

- 1) **monitoraggio continuo dei rischi e del contesto della loro gestione per eventuali loro cambiamenti;**
- 2) **monitoraggio continuo di possibili nuovi rischi durante l'intero ciclo di vita.**

Il monitoraggio dei rischi avviene principalmente in sede di meeting interno a seguito di analisi retrospettive.

4.5 Processo di Gestione della Configurazione

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.5

4.5.1 Scopo

Lo scopo della Gestione della Configurazione è gestire e controllare gli elementi del sistema e le configurazioni durante il ciclo di vita. Lo scopo principale è tracciare e coordinare le procedure necessarie alla modifica della documentazione_G e del codice sorgente.

4.5.2 Risultati

Come risultato dell'implementazione riuscita del processo di Gestione della Configurazione:

- vengono identificati e gestiti gli elementi che richiedono la gestione della configurazione;
- vengono stabilite le linee base di configurazione;
- sono controllate le modifiche agli elementi sotto gestione della configurazione;
- sono disponibili informazioni sullo stato della configurazione;
- le release dei documenti e del software sono controllate e approvate.

4.5.3 Attività

4.5.3.1 Versionamento_G

4.5.3.1.1 Generalità

Il versionamento_G è un processo che permette di tenere traccia delle modifiche effettuate su un prodotto software o documentale. Per ogni modifica viene creata una nuova versione_G del prodotto, che viene identificata da un numero di versione_G. Il numero di versione_G è composto da tre cifre separate da un punto, e segue la convenzione seguente:

X.Y.Z

dove:

- X: indica il numero di versione_G principale;
- Y: indica il numero di versione_G secondaria;
- Z: indica il numero di versione_G di revisione_G e correzione.

Per il versionamento_G di qualsiasi prodotto del progetto è essenziale osservare i principi elencati di seguito:

- 1) l'aggiornamento di una delle cifre del numero di versione_G azzerà le cifre di rilevanza inferiore;
- 2) un normale numero di versione_G deve avere la forma X.Y.Z, dove X, Y e Z sono interi non negativi e non preceduti da zeri;
- 3) numeri di versione_G con X pari a 0 indicano prodotti in lavorazione, da non considerarsi pronti al rilascio;
- 4) dopo il rilascio, il contenuto della versione_G non deve essere modificato. Qualsiasi modifica successiva al rilascio deve causare un cambio nel numero di versione_G.

4.5.3.1.1.1 Documentazione_G

L'aggiornamento del numero di versione_G per la documentazione_G deve attenersi alle seguenti regole:

- X: deve essere aggiornato al cambiamento della struttura del documento. Riguarda dunque cambiamenti di organizzazione del documento, dei suoi paragrafi e della presentazione delle informazioni, nonché cambiamento dei parametri necessari nel template dei documenti;
- Y: deve essere aggiornato all'aggiunta o alla rimozione di paragrafi;
- Z: deve essere aggiornato a seguito di cambiamenti minimi o correzioni ortografiche.

Il processo di aggiornamento della versione_G è reso automatico, come stabilito nella sezione dedicata a GitHub Actions_G (Sezione 4.5.3.1.3.2).

4.5.3.1.1.2 Software

L'aggiornamento del numero di versione_G per il software deve attenersi alle seguenti regole:

- X: deve essere aggiornato solo al raggiungimento di una versione_G del prodotto che abbia un set di funzionalità il più completo possibile. Inizialmente è dunque posto a 0, e passerà a 1 solamente al momento in cui il prodotto sarà classificabile come MVP_G;
- Y: deve essere aggiornato ogniqualvolta viene introdotta una nuova funzione nel codice o una nuova funzionalità al prodotto;
- Z: deve essere aggiornato ad ogni correzione di bug_G o cambiamento minore. Alcuni esempi di cambiamenti minori sono:
 - correzioni ortografiche:
 - nei nomi di variabili;
 - nei nomi di parametri;
 - nei nomi di funzioni o metodi.
 - cambiamento nel valore di parametri:
 - cambiamento di colore di un elemento grafico;
 - cambiamento di un valore numerico;

- cambiamento di un valore letterale.

4.5.3.1.2 Tracciamento modifiche

Il tracciamento delle modifiche avviene per mezzo di automazioni che permettono di identificare:

- versione_G del documento modificato;
- data di modifica (gg-mm-aaaa, ddd);
- numero della pull request_G di riferimento;
- titolo della pull request_G di riferimento;
- autore_G della modifica;
- revisore incaricato.

Tali informazioni sono salvate in un file CSV, unico per ogni documento. Questo file, denominato *log.csv*, è salvato nella cartella dedicata al documento a cui si riferisce, e viene generato automaticamente da una GitHub_G Action invocata all'apertura, riapertura, sincronizzazione e chiusura di una pull request_G. Maggiori dettagli al paragrafo dedicato (Sezione 4.5.3.1.3.2).

Ogni documento, nella sezione direttamente sottostante all'indice, mostra in formato tabellare le informazioni relative al tracciamento delle modifiche, leggendo le informazioni dal file *log.csv*.

Ver.	Data	PR _G	Titolo	Redattore _G	Verificatore
1.0.0	11-12-2023, Mon	90	DOC-123 Redazione paragrafo	Riccardo Carraro	Mattia Todesco

Tabella 2: Esempio tracciamento modifiche

4.5.3.1.3 Tecnologie adoperate

4.5.3.1.3.1 GitHub_G

Lo strumento di versionamento_G scelto dal gruppo è GitHub_G. Il gruppo *Error_418* ha creato un'organizzazione omonima su GitHub_G in modo da gestire e separare il lavoro in più repository_G pensate per scopi e contenuti diversi:

- Documenti: documentazione_G;
- WMS3: codice sorgente;
- PoC_G: Proof of Concept_G.

Documenti è la repository_G dedicata alla documentazione_G prodotta, la quale possiede tre branch_G principali:

- **main**: contiene i file pdf dei documenti prodotti solamente in seguito ad un processo di review_G con esito positivo;
- **src**: contiene i file sorgente dei documenti prodotti, file di configurazione e di supporto;
- **website**: contiene i file sorgente del sito web che espone la documentazione_G del gruppo.

Documenti è organizzata in modo da suddividere la documentazione_G necessaria alle revisioni esterne che il gruppo dovrà affrontare:

- **RTB_G**: contiene i file necessari alla Requirements and Technology Baseline_G:
 - **Documentazione_G esterna**: contiene i documenti ad uso esterno:
 - **Verbali**: contiene i verbali delle riunioni esterne;

- **doc_file_name:**
 - *doc_file_name.typ*: file sorgente del documento;
 - *log.csv*: registro delle modifiche_G associato al documento.
- **Documentazione_G interna:** contiene i documenti ad uso interno:
 - **Verbali:** contiene i verbali delle riunioni interne;
 - **doc_file_name:**
 - *doc_file_name.typ*: file sorgente del documento;
 - *log.csv*: registro delle modifiche_G associato al documento.
- **PB_G:** contiene i file necessari alla Product Baseline_G. È organizzata allo stesso modo della cartella dedicata alla RTB_G.

Al fine di garantire uno svolgimento delle attività in parallelo, la strategia utilizzata dal gruppo durante lo sviluppo è il *featuring branching*. È presente un *branch_G* per le release e un *branch_G* per lo sviluppo dal quale vengono creati dei *branch_G* per ogni nuova funzionalità o modifica da apportare. Questi ultimi vengono identificati dal codice **DOC-XXX**, dove **XXX** è il numero del relativo *task_G* su Jira_G. I *branch_G* di *feature_G* vengono integrati tramite *pull request_G*.

WMS3 è la *repository_G* che contiene il codice sorgente del prodotto software sviluppato dal gruppo. È presente un *branch_G* principale, **dev**, in cui si trova il codice sorgente relativo all'ultima versione_G del prodotto, reperibile anche nella sezione Releases presente sulla destra dell'interfaccia di GitHub_G.

La *repository_G* è organizzata in due sottocartelle principali, da cui si diramano diverse cartelle secondarie:

- **db_G:** contiene il file *SQL_G* utile alla creazione e popolamento del *database_G* di supporto;
- **web:** contiene il codice sorgente di frontend e backend, e si dirama in:
 - **__test__:** contiene i test del software, in particolare:
 - **DataMapperTest:** test riguardanti il pattern Data Mapper;
 - **Integration Test:** test di integrazione_G;
 - **Unit Test:** test di unità_G delle classi del modello.
 - **app:** file riguardanti la pagina principale e lo stile;
 - **components:** componenti utilizzati nel prodotto, in particolare:
 - **custom:** componenti personalizzati, quali:
 - form di creazione dell'ambiente 3D;
 - pannelli per la visualizzazione delle impostazioni e delle varie liste di elementi del prodotto.
 - **providers:** file relativi all'implementazione del pattern Provider;
 - **ThreeComponents:** file relativi agli elementi 3D;
 - **ui:** file relativi agli elementi dell'interfaccia grafica.
 - **dataMapper:** file relativi all'implementazione del pattern Data Mapper;
 - **dataRepository:** file relativi all'implementazione del pattern Repository_G;
 - **lib:** contiene un file riguardante il merge tra più classi del framework Tailwind CSS;
 - **model:** file relativi alle classi del modello;
 - **node_G_modules:** cartella generata automaticamente da **npm** (Node_G Package Manager) utile alla gestione delle dipendenze di Node.js_G;
 - **pages:** API_G prodotte dal gruppo;

- **public:** file SVG_G per la creazione del magazzino_G e una sottocartella di icone;
- **ServerActions:** file relativi all’implementazione delle Server Actions utilizzate;
- **Strategy:** file relativi all’implementazione del pattern Strategy.

Anche in questa repository_G, come per Documenti, viene applicato il *feature_G branching*, utilizzando *dev* come branch_G di appoggio per l’apertura dei branch_G di feature_G, identificati dal codice *WMS-XXX*, dove XXX è il numero del relativo task_G su Jira_G.

4.5.3.1.3.2 GitHub Actions_G

L’intero processo di versionamento_G è accompagnato da una serie di automazioni, realizzate tramite GitHub Actions_G, che sollevano i componenti del gruppo dall’onere manuale di attività come la compilazione dei documenti, l’aggiornamento del registro delle modifiche_G (file *log.csv*) e la pubblicazione dei documenti dopo la verifica.

Workflow delle automazioni:

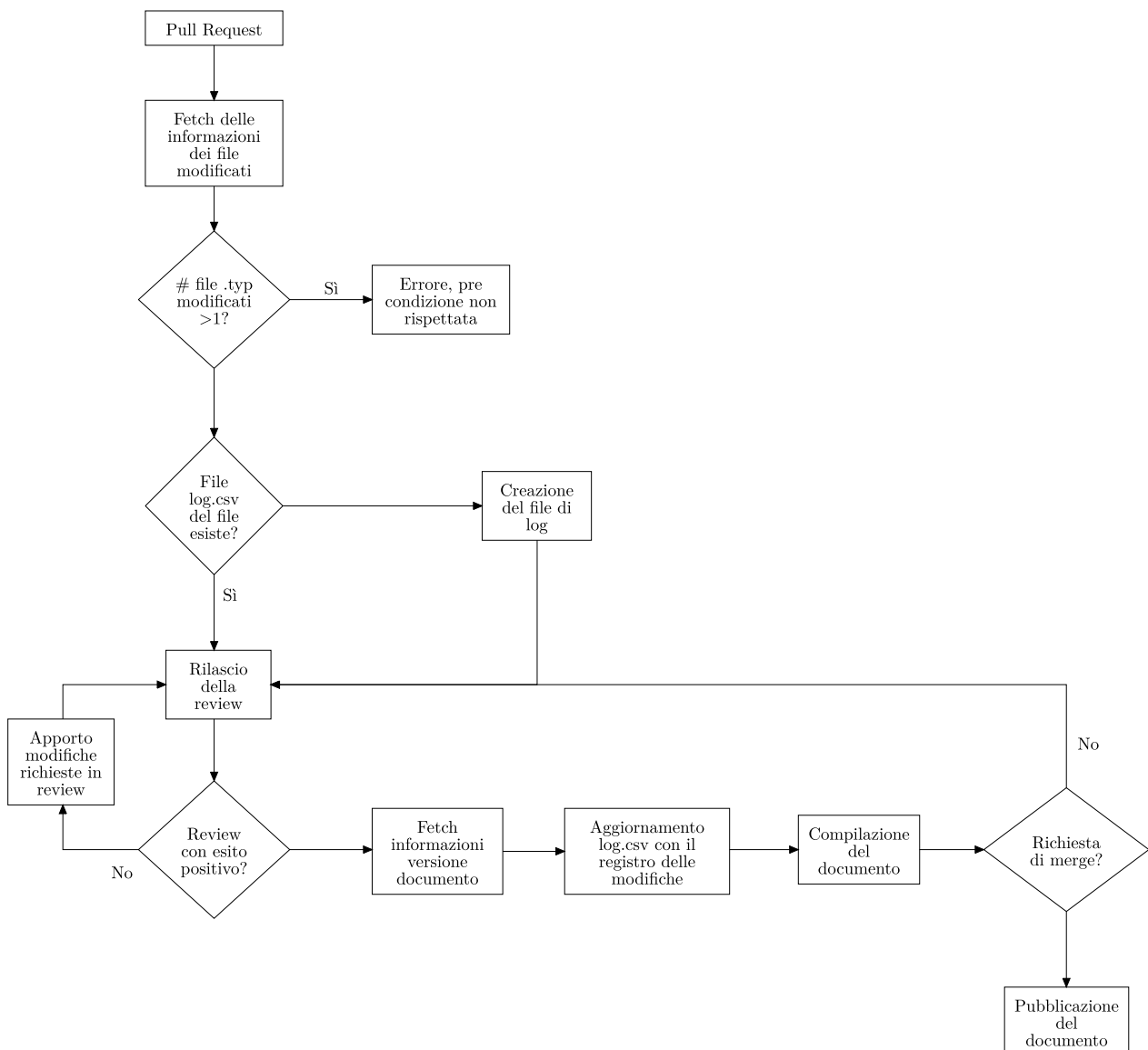


Figura 1: Workflow delle automazioni

Alla creazione della pull request_G si avvia il workflow per la compilazione e la registrazione delle modifiche avvenute. Prima di procedere è necessario inserire informazioni essenziali ai fini di ottenere maggiore chiarezza e tracciabilità nel messaggio di pull request_G, quali:

- titolo conforme, contenente il nome del task_G di riferimento su Jira_G legata alla pull request_G, nel formato `DOC-XXX titolo`;
- identificativo di almeno un verificatore;
- eventuali note aggiuntive.

Il workflow è composto dai seguenti passaggi:

- 1) **fetch delle informazioni dei file modificati**: vengono recuperate le informazioni relative ai file modificati nella pull request_G, quali:
 - nome del file;
 - path del file.
- 2) **controllo del numero di file modificati**: se il numero di file modificati è maggiore di 1, il workflow termina con un errore;
- 3) **controllo dell'esistenza del file *log.csv***: se il file non esiste, viene creato (sinonimo di creazione del documento);
- 4) **rilascio della review_G**: il verificatore si occupa di controllare il documento e rilasciare la review_G, segnando i cambiamenti richiesti;
- 5) **richiesta di una nuova review_G per verificare che i cambiamenti apportati siano corretti**;
- 6) **nel momento in cui la review_G termina con esito positivo, si procede al recupero della versione_G corrente del documento**:
 - se non esiste il corrispettivo pdf nel branch_G main, allora il documento non era mai stato pubblicato, pertanto la sua versione_G di partenza sarà fissata a 1.0.0;
 - se esiste il corrispettivo pdf nel branch_G main, essendo la versione_G contenuta nel nome del file, si procede al recupero della versione_G corrente del documento, modificando la versione_G X.Y.Z in base all'analisi del documento mediante uno script python;
- 7) **aggiornamento del file *log.csv***: il file di log viene aggiornato con le informazioni relative alla modifica effettuata: questo passaggio, avvenendo solamente a seguito di review_G positiva, permette di garantire che vengano segnate solamente le modifiche che hanno superato il processo di verifica;
- 8) **compilazione del documento**: aggiornato il file *log.csv* e recuperato il numero di versione_G, il documento è pronto per essere compilato, mostrando numero di versione_G e registro delle modifiche_G aggiornati;
- 9) **pubblicazione del documento**: terminati i workflow precedenti, se si avvia la procedura di merge a seguito del processo di verifica, il documento pdf generato dalla compilazione viene pubblicato nel ramo main della repository_G;
- 10) **merge non confermato**: qualora a termine della compilazione del documento non venisse confermato il merge da parte del verificatore, significa che è stato individuato un ulteriore errore o correzione da dover apportare al documento prima della sua pubblicazione sul ramo main del

repository_G. In questa circostanza sarà dunque necessario rilasciare un'ulteriore review_G. L'esecuzione riprende dal punto 4.

L'azione manuale si riduce solamente al rilascio di review_G e conferma di merge, mentre tutte le altre attività vengono automatizzate. All'approvazione della pull request_G, e alla conseguente chiusura del branch_G, un'ulteriore automazione integrata su Jira_G, permette di aggiornare in automatico lo stato del task_G collegato alla pull request_G, portandolo allo stato di "Completato".

4.5.3.1.3.3 Typst_G

Il gruppo utilizza Typst_G come strumento di scrittura e compilazione dei documenti.

Al fine di dare una struttura comune ai documenti si è creato un file *template.typ* parametrizzato, sfruttando la possibilità di produrre un file pdf compilando insieme più file Typst_G. Questo file contiene le impostazioni di base per la creazione di un documento:

- **title**: titolo del documento;
- **subTitle**: sottotitolo del documento;
- **docType**: tipologia del documento (Verbale_G, Documento);
- **date**: data di creazione del documento;
- **externalParticipants**: partecipanti esterni al gruppo;
- **authors**: autori del documento;
- **reviewers**: revisori del documento;
- **missingMembers**: membri assenti durante i meeting;
- **location**: luogo di incontro;
- **timeStart**: ora di inizio incontro;
- **timeEnd**: ora di fine incontro;
- **showLog**: flag che indica se mostrare il tracciamento delle modifiche;
- **showIndex**: flag che indica se mostrare l'indice;
- **isExternalUse**: flag che indica se il documento è per uso esterno.

Al momento della creazione di un nuovo documento sarà sufficiente importare il modello e specificare i parametri sopra elencati.

Al fine di semplificare la procedura di creazione di un documento, è stato condiviso un documento di testo denominato *quickstart.txt* che contiene la configurazione base per la stesura dei documenti.

4.5.3.2 Tracciamento dei task_G e amministrazione dello stato di configurazione

4.5.3.2.1 Jira_G

Jira_G è lo strumento centrale per la gestione e la tracciabilità dei task_G assegnati ai membri del gruppo. L'integrazione con GitHub_G permette a Jira_G di lavorare e apportare modifiche direttamente al repository_G del gruppo, permettendo la creazione, gestione e chiusura di branch_G e al conseguente aggiornamento dello stato dei task_G. Ogni task_G è identificato da un codice univoco incrementale nel formato DOC-XXX, dove XXX è un numero positivo sequenziale, che permette di identificarlo. Ogni task_G è caratterizzato da:

- codice identificativo DOC-XXX generato automaticamente da Jira_G;
- titolo;
- descrizione (opzionale);

- stato (“Da completare”, “In corso”, “In verifica”, “Completato”);
- assegnatario;
- story_G point (stima del carico di lavoro);
- epic_G story_G (milestone_G) di riferimento.

Nel processo di versionamento $_G$ e di tracciamento delle modifiche, Jira_G ricopre un ruolo fondamentale, grazie anche alla sua integrazione con GitHub_G : nel momento in cui si intende avviare un task $_G$, è necessario seguire i seguenti passaggi:

- aprire il task $_G$ su Jira_G ;
- selezionare l’opzione di creazione di un branch_G dedicato al task $_G$ (integrazione con GitHub_G);
- selezionare la repository $_G$ e il branch_G da cui creare il nuovo branch_G .

A questo punto, il task $_G$ si aggiornerà nello stato “In corso” e verrà aperto il relativo branch_G . Terminato il task $_G$ ed effettuata la pull request $_G$, lo stato del ticket $_G$ passerà automatica a “In verifica”. Superato il processo di verifica, Jira_G provvederà ad aggiornare lo stato del task $_G$ in “Completato”.

4.5.3.2.1.1 Backlog $_G$

Ogni task $_G$ da svolgere è segnato all’interno del backlog $_G$ di Jira_G . Durante la pianificazione dello sprint $_G$, si definisce lo sprint $_G$ backlog $_G$, il sottoinsieme di attività provenienti dal backlog $_G$ che si intendono portare a termine entro la conclusione dello sprint $_G$. A differenza dello sprint $_G$ backlog $_G$ definito durante la pianificazione, il backlog $_G$ viene espanso man mano che si riscontrano nuovi task $_G$ necessari o a seguito di decisioni prese durante le riunioni interne o esterne.

4.5.3.2.1.2 Board

Le board di Jira_G permettono, similmente allo sprint $_G$ backlog $_G$, di avere una visione d’insieme delle attività da svolgere, ma con un approccio più visuale e intuitivo. I task $_G$ sono organizzati in quattro colonne, rappresentanti lo stato:

- **da completare:** non ancora avviati, ovvero non esiste il branch_G dedicato;
- **in corso:** in fase di svolgimento, ovvero branch_G dedicato al task $_G$ creato;
- **in verifica:** in fase di review $_G$, dopo l’apertura di una pull request $_G$;
- **completato:** task $_G$ concluso, ovvero branch_G dedicato chiuso a seguito di merge sul ramo principale.

4.5.3.2.1.3 Timeline

La timeline di Jira_G permette di avere una visione delle attività incentrata sulle tempistiche e le relazioni tra i task $_G$. Permette inoltre di mostrare il grafico di Gantt $_G$ delle attività evidenziando i rapporti di dipendenza tra i task $_G$ e stabilendo le scadenze necessarie per il loro svolgimento.

4.5.3.2.1.4 Grafici

Jira_G offre la possibilità di produrre grafici e report relativi all’avanzamento e alla tracciabilità dei task $_G$. Tali strumenti permettono di avere delle metriche di valutazione dell’andamento del progetto e di individuare eventuali criticità. Il gruppo utilizza come metrica principale il burndown $_G$ chart, che permette di avere una visione dell’avanzamento delle attività in base al tempo, basato sugli story_G point di ogni attività.

4.5.3.3 Controllo delle release

Il controllo delle release viene gestito tramite il meccanismo di pull request_G di GitHub_G. Prima di integrare i nuovi cambiamenti, viene aperta una pull request_G dall'assegnatario del task_G. La pull request_G deve avere un titolo valido (come descritto nel paragrafo dedicato Sezione 4.5.3.1.3.2) e deve essere designato almeno un reviewer. Di norma il reviewer di base è il Verificatore, che svolge una supervisione sulla correttezza sintattica e semantica dei contenuti. Nel caso in cui ci sia bisogno di una figura con delle competenze specifiche per quanto riguarda la semantica e il contenuto del materiale da revisionare, il Verificatore può essere affiancato da altri membri del gruppo. Per ogni richiesta di modifiche da apportare vengono aperte delle conversation, in cui è possibile evidenziare le linee del documento che hanno bisogno di cambiamenti, oltre a stabilire un canale di comunicazione fra assegnatario e Verificatore. Il processo di verifica del documento è accompagnato dall'esecuzione di GitHub Actions_G che si occupano di automatizzare l'aggiornamento del file *log.csv* con i dati relativi alla modifica apportata, e la compilazione e pubblicazione del documento nel ramo main del repository_G. In questo modo si assicura che ogni documento presente nel ramo principale sia prima stato sottoposto ad un processo di verifica. Si può procedere alla chiusura della pull request_G e con l'operazione di merge solo nel caso in cui tutte le conversation siano state risolte e siano stati applicati i cambiamenti richiesti in fase di review_G. Solo a seguito del merge della pull request_G, il task_G collegato presente in Jira_G può essere definito concluso e il relativo branch_G viene chiuso in modo automatico.

4.6 Processo di Gestione delle Informazioni

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.6

4.6.1 Scopo

Lo scopo del processo di Gestione delle Informazioni è generare, ottenere, confermare, trasformare, conservare, recuperare e distribuire le informazioni e la relativa documentazione_G agli stakeholder interessati, garantendone chiarezza, completezza, consistenza, tracciabilità e presentazione.

Le informazioni possono essere tecniche, di progetto, organizzative e di accordo.

4.6.2 Informazioni gestite

Le informazioni gestite dal gruppo sono:

- documentazione_G:
 - *Piano di Progetto*_G
 - *Norme di Progetto*_G
 - *Piano di Qualifica*_G
 - *Analisi dei Requisiti*_G
 - *Glossario*
 - Verballi.
- codice sorgente:
 - Proof of Concept_G (PoC_G);
 - Minimum Viable Product_G (MVP_G).

Codice sorgente e documenti sono creati, organizzati, aggiornati, versionati e distribuiti all'interno dei repository_G del gruppo.

4.6.3 Documentazione_G

4.6.3.1 Struttura dei documenti

Ogni documento segue una struttura standard, stabilita nel template *template.typ*.

Questo paragrafo non tratta della struttura dei verbali. Per le norme sulla struttura dei verbali si rimanda alla Sezione 4.6.3.2.

I documenti pertanto sono così strutturati:

- 1) **Cover page**: la cover page è la pagina iniziale del documento. Essa contiene le seguenti informazioni:
 - nome del gruppo;
 - link all'organizzazione GitHub_G;
 - indirizzo e-mail del gruppo;
 - logo;
 - titolo del documento;
 - sottotitolo del documento;
 - versione_G;
 - stato;
 - tipo di uso: interno od esterno;
 - responsabile del gruppo;
 - redattori;
 - verificatori;
 - destinatari;
 - partecipanti esterni (se presenti).
- 2) **Registro delle modifiche**_G: sezione successiva alla cover page. Maggiori dettagli sono disponibili alla sezione dedicata (Sezione 4.5.3.1.2);
- 3) **Indici**: Sono presenti tre tipologie di indici:
 - indice del contenuto: indice sempre presente che rappresenta i paragrafi del documento;
 - indice delle tabelle: indice presente solo se sono presenti tabelle nel documento;
 - indice delle figure: indice presente solo se sono presenti figure nel documento.
- 4) **Contenuto del file**: sezione successiva agli indici. Rappresenta il corpo del documento, suddiviso in paragrafi.

4.6.3.2 Struttura dei verbali

Al fine di rendicontare argomenti, decisioni, aggiornamenti, problematiche e attività oggetto di discussione durante i meeting interni ed esterni, il gruppo raccoglie tali informazioni all'interno di verbali redatti al termine dei meeting. La struttura dei verbali è la seguente:

- 1) **cover page** (Sezione 4.6.3.1);
- 2) **"Informazioni generali"**:
 - luogo: il luogo reale o digitale dove il gruppo insieme ad eventuali partecipanti esterni svolge il meeting;
 - data e ora nel formato (gg-mm-aaaa, hh:mm hh:mm);
 - partecipanti;

- assenti;
 - partecipanti esterni al gruppo (se presenti).
- 3) “**Ordine del giorno**”: elenco degli argomenti trattati durante la riunione;
- 4) “**Valutazione del progresso generale**” descrive l’analisi sul lavoro svolto durante lo sprint_G precedente. Questo paragrafo è così definito:
- valutazione sulla qualità del lavoro in merito alla velocità di avanzamento, evidenziando eventuali rallentamenti o meriti riscontrati.
 - sottoparagrafi per ogni epic_G su cui si è lavorato in cui viene riportato una descrizione sul lavoro svolto, se i lavori svolti sono molti e diversi riportarli in forma di elenco puntato.
- 5) “**Analisi di retrospettiva**”, suddiviso in:
- lista di valori calcolati dalle metriche presenti nel *Piano di Qualifica*_G riguardo allo sprint_G, nello specifico riportare:
 - CPI_G: valore precedente e valore attuale;
 - EAC_G: valore precedente e valore attuale;
 - rapporto tra SEV_G e SPV_G.
 - sottoparagrafo “**Keep doing**”, riporta i meriti riscontrati dal gruppo;
 - sottoparagrafo “**Improvements**”, riporta le criticità riscontrate dal gruppo etichettate con un codice identificativo.

I codici sono così composti:

- **P**, lettera indicante una Problematica;
- **O/T**, a seconda se la problematica sia di origine organizzativa o tecnologica;
- **N**, intero naturale positivo che incrementa ad ogni nuova problematica di tipo organizzativo o tecnologico.

Quindi inserire la tabella che associa ogni criticità ad una risoluzione presente nel *Piano di Progetto*_G. La tabella deve riportare:

- **ID risoluzione**;
- **Titolo risoluzione**;
- **Criticità affrontate**.

- 6) eventuali ulteriori argomenti organizzati in paragrafi indipendenti;
- 7) “**Pianificazione**”, riporta la tabella contenente le task_G programmate per lo Sprint_G successivo. Questa tabella è automaticamente generata da un foglio di calcolo elettronico condiviso realizzato in Google Sheets_G (maggiori dettagli in merito reperibili al paragrafo Sezione 4.8.3.1.2) e disponibile nel Google Drive_G del gruppo.

4.6.4 Stile e convenzioni

Al fine di uniformare e conformare i prodotti del progetto, il gruppo ha stabilito delle convenzioni stilistiche e di scrittura da rispettare durante la stesura dei documenti e del codice. L’obiettivo è perseguire:

- chiarezza;
- leggibilità;
- manutenibilità.

4.6.4.1 Convenzioni stilistiche globali

Convenzioni stilistiche valide sia per i prodotti documentali che software.

4.6.4.1.1 Nomi dei documenti

Ogni parola dei titoli dei documenti deve iniziare con la lettera maiuscola, ad eccezione delle preposizioni e degli articoli.

I verbali avranno come titolo la data del verbale_G nel formato *yyyy-mm-dd*. Se si fa riferimento al contenuto di un documento, dopo il rispettivo titolo riportare anche la versione_G nel formato *_vX.Y.Z*.

Esempio: *Norme di Progetto_G v1.30.1*.

4.6.4.1.2 Formato data

All'interno del documento, le date seguiranno il formato locale *dd/mm/yyyy*, mentre all'interno dei nomi dei file e dei commit_G di GitHub_G, il formato utilizzato sarà *yyyy-mm-dd*, dove:

- **dd**: numero del giorno con due cifre;
- **mm**: numero del mese con due cifre;
- **yyyy**: numero dell'anno con quattro cifre.

4.6.4.2 Convenzioni stilistiche documentali

Convenzioni stilistiche specifiche per i prodotti documentali.

4.6.4.2.1 TODO

Per indicare sezioni del documento da completare, il gruppo ha deciso di utilizzare il termine TODO, che verrà in automatico mostrato in rosso e riquadrato, riportando il messaggio “*Riferimento assente*”.

Il risultato è il seguente: Riferimento assente.

Questo permette di individuare facilmente le parti del documento da completare.

4.6.4.2.2 Corsivo, grassetto, maiuscole e monospace

Corsivo:

- citazioni;
- formati;
- nomi di practice;
- nomi propri di dominio.

Grassetto:

- titoli;
- parole chiave e significative;
- termini iniziali di elenchi puntati che necessitano spiegazione.

MAIUSCOLO:

- acronimi;
- nomi propri;
- nomi strumenti e tecnologie;
- iniziale nomi ruoli;
- iniziale parole nei nomi documenti ad eccezione di preposizioni e articoli. Riferimento nomi file disponibile alla Sezione 4.6.4.1.1.

Monospace:

- nome di un file (Riferimento nomi file disponibile alla Sezione 4.6.4.1.1);
- parametri;
- porzioni di codice.

4.6.4.2.3 Elenchi

- si utilizzano elenchi numerati se gli elementi mostrati richiedono un ordine (es. ordine delle sezioni);
- si utilizzano elenchi non numerati se gli elementi mostrati non richiedono un ordine (es. lista di attività);
- al termine di ogni punto dell'elenco viene posto “;” ad eccezione dell'ultimo elemento a cui viene posto “.”.

4.6.4.2.4 Glossario

Tutte le occorrenze dei termini contenuti nel glossario sono evidenziati con una G in corsivo a pedice.

4.6.4.2.5 Variabili Typst_G per termini ricorrenti

Al fine di uniformare termini ricorrenti, vengono predisposte variabili utilizzabili nella redazione dei documenti. I termini in considerazione sono relativi a:

- nomi di documenti
- stakeholder;
- documenti e la versione_G a cui si fa riferimento;
- ultima data di accesso ad un link ad una risorsa web esterna.

Di seguito viene riportata la tabella delle variabili e la loro rappresentazione all'interno del documento compilato:

Variabile	Risultato
\#err418	<i>Error_418</i>
\#cardin	Professor Cardin
\#vardanega	Professor Vardanega
\#adr _G	<i>Analisi dei Requisiti_G</i>
\#adr _v	<i>Analisi dei Requisiti_G v2.0.1</i>
\#ris	<i>Analisi dei Rischi_G</i>
\#ris _v	<i>Analisi dei Rischi_G v2.0.1</i>
\#glo	<i>Glossario</i>
\#glo _v	<i>Glossario v1.3.0</i>
\#ndp _G	<i>Norme di Progetto_G</i>
\#ndp _v	<i>Norme di Progetto_G v1.30.1</i>
\#pdp _G	<i>Piano di Progetto_G</i>
\#pdp _v	<i>Piano di Progetto_G v1.15.0</i>
\#pdq _G	<i>Piano di Qualifica_G</i>
\#pdq _v	<i>Piano di Qualifica_G v1.5.0</i>
\#lastVisitedOn(day, month, year)	<i>(ultimo accesso 01/01/2024)</i>

Tabella 3: Variabili Typst_G per riferirsi a termini ricorrenti

4.6.5 Distribuzione delle informazioni

Il gruppo condivide il materiale prodotto all'interno di un repository_G dedicato reperibile al link:

<https://github.com/Error-418-SWE/Documenti>

Maggiori dettagli in merito all'organizzazione della repository_G sono reperibili qui: Sezione 4.5.3.1.3.1.

4.7 Processo di Misurazione

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.7

4.7.1 Scopo

Lo scopo del processo di misurazione è raccogliere, analizzare e riportare dati e informazioni, al fine di dimostrare la qualità di prodotti, servizi e processi.

Questo processo viene svolto affinché il gruppo sia in grado di prendere le corrette decisioni per gestire al meglio il raggiungimento dei propri obiettivi.

Le misure devono soddisfare determinate caratteristiche di qualità, ovvero devono essere:

- 1) verificabili;
- 2) significative;
- 3) attuabili;
- 4) tempestive;
- 5) economicamente vantaggiose.

Definiamo il concetto di “qualità” come segue: insieme delle caratteristiche di un'entità, che ne determinano la capacità di soddisfare esigenze sia esplicite che implicite.

4.7.2 Risultati

A seguito dell'implementazione efficace del processo di misurazione:

- vengono identificate le esigenze informative;
- viene identificato e sviluppato un insieme appropriato di misure basato sulle esigenze informative;
- i dati necessari vengono raccolti, verificati e archiviati;
- i dati vengono analizzati e i risultati interpretati;
- gli elementi informativi forniscono informazioni oggettive per poter prendere decisioni concrete.

I risultati sono contenuti nel documento *Piano di Qualifica_G v1.5.0*.

4.7.3 Attività

Il gruppo deve implementare le seguenti attività in conformità con le politiche e le procedure applicabili al processo di misurazione definito nel *Piano di Qualifica_G v1.5.0*:

- 1) prepararsi per la misurazione:
 - definire la strategia di misurazione per i processi primari. La strategia scelta si compone di metriche mirate alla valutazione dei processi primari, quali:
 - a) BAC (Budget at Completion);
 - b) PV (Planned Value), che si estende in:
 - SPV_G (Sprint Planned Value_G);

- PPV_G (Project Planned Value_G);
 - c) AC_G (Actual Cost_G), che si estende in:
 - SAC_G (Sprint Actual Cost_G);
 - PAC_G (Project Actual Cost_G);
 - d) EV_G (Earned Value_G), che si estende in:
 - SEV_G (Sprint Earned Value_G);
 - PEV_G (Project Earned Value_G);
 - e) CPI_G (Cost Performance Index_G);
 - f) EAC_G (Estimated at Completion).
- definire la strategia di misurazione per i processi di supporto. La strategia scelta si compone di una serie di parametri e metriche che permettono di valutare la qualità dei processi di supporto:
 - errori ortografici_G;
 - percentuale metriche soddisfatte_G.
 - definire la strategia di misurazione per la qualità del prodotto. La strategia scelta si compone di metriche mirate alla valutazione della qualità del prodotto, quali:
 - a) MRC (Mandatory Requirements Coverage);
 - b) DRC (Desiderable Requirements Coverage);
 - c) ORC (Optional Requirements Coverage);
 - d) ART (Average Response Time);
 - e) LT (Learning Time);
 - f) EOU (Ease of Use);
 - g) CC (Cyclomatic Complexity);
 - h) CL (Coupling Level);
 - i) RC (Responsability Count);
 - j) MPN (Method Parameters Number);
 - k) FD (Failure Density);
 - l) SBV (Supported Browser_G Version_G).
 - descrivere le caratteristiche del gruppo rilevanti per la misurazione, come obiettivi aziendali e obiettivi tecnici:
 - richieste del Proponente;
 - requisiti individuati.
 - definire procedure di raccolta, analisi, accesso e reportistica dei dati:
 - *Piano di Qualifica_G*: definisce i criteri e le modalità di misurazione e reportistica dei dati misurati.
 - definire criteri per valutare gli elementi informativi e il processo di misurazione:
 - *Piano di Qualifica_G*: definisce i valori ottimali e accettabili della strategia di misurazione adottata a cui tutti i documenti prodotti devono conformarsi.
 - identificare e pianificare le azioni da intraprendere per i casi in cui i parametri di misurazione di qualità non vengano rispettati.

2) eseguire la misurazione:

- integrare procedure manuali o automatizzate per la generazione, raccolta, analisi e reportistica dei dati nei processi pertinenti:
 - controllo manuale di conformità dei documenti prodotti ai parametri individuati per i processi primari;
 - controllo manuale di conformità dei documenti prodotti ai parametri individuati per i processi di supporto;
 - revisione_G esterna e manuale dei documenti prodotti per il controllo di leggibilità e di eventuali errori ortografici_G.
- raccogliere, archiviare e verificare i dati:
 - i dati prodotti vengono salvati e analizzati al fine di perorare una strategia di approccio ed eventuali modifiche da effettuare.
- registrare i risultati e comunicarli agli stakeholder:
 - vengono riferiti periodicamente al Proponente gli avanzamenti e la conformità del progetto alle richieste concordate.

Maggiori dettagli in merito alla definizione, al calcolo e l'analisi delle metriche sono reperibili all'interno del documento *Piano di Qualifica_G v1.5.0*.

4.8 Processo di Controllo della Qualità

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.3.8

4.8.1 Scopo

La International Software Testing_G Qualifications Board (ISTQB) definisce il processo di Controllo della Qualità come “insieme di attività incentrate sul garantire che i requisiti di qualità saranno soddisfatti”.

Il processo di Controllo della Qualità ha come obiettivo principale quello di garantire l'applicazione efficace del processo di Gestione della Qualità da parte del gruppo.

Il suo focus fondamentale è fornire la sicurezza e la certezza che i requisiti di qualità stabiliti saranno pienamente soddisfatti. Questo si ottiene mediante un'analisi proattiva attenta dei processi e dei risultati che caratterizzano il ciclo di vita del progetto.

L'obiettivo è assicurare che il prodotto risponda alle aspettative in termini di qualità, e che ciò avvenga in piena conformità con le linee guida, le politiche e le procedure precedentemente definite. In altre parole, la finalità ultima è garantire che la qualità desiderata sia intrinseca al prodotto finale e che ogni fase del processo segua gli standard e le disposizioni stabilite dal gruppo e dal progetto.

4.8.2 Risultati

L'implementazione riuscita del processo di Controllo della Qualità produce i seguenti risultati:

- sono definite e implementate le procedure di Controllo della Qualità del progetto;
- vengono definiti criteri e metodi per la valutazione di Controllo della Qualità;
- vengono eseguite valutazioni dei prodotti, servizi e processi del progetto;
- vengono forniti agli stakeholder i risultati delle valutazioni;
- vengono trattati i problemi emersi durante il periodo di sviluppo.

4.8.3 Attività

Le seguenti attività devono essere implementate in conformità con le politiche e le procedure del gruppo:

1) **prepararsi per il Controllo della Qualità:**

- **definire una strategia di Controllo della Qualità.** Essa consiste in:
 - a) **ruoli, responsabilità e autorità definite:**
 - i ruoli e i compiti di ciascun membro sono definiti e non ambigui;
 - ogni individuo fisico viene informato a scadenza bisettimanale dei propri compiti e delle proprie responsabilità per quel periodo di tempo.
 - b) **criteri di valutazione definiti per processi, prodotti e servizi:**
 - il valore delle metriche di controllo dei processi primari deve essere accettabile;
 - il valore delle metriche di controllo dei processi di supporto deve essere accettabile;
 - le tempistiche da rispettare devono essere definite;
 - i requisiti funzionali devono essere definiti;
 - i requisiti non funzionali devono essere definiti.
 - c) **attività di verifica, convalida, monitoraggio, misurazione, revisione_G per i prodotti o servizi:**
 - assegnazione di un Verificatore con il compito di monitorare e testare la qualità del materiale prodotto;
 - il Verificatore ha il compito di segnalare eventuali incongruenze con le metriche di qualità al redattore_G.
 - d) **risoluzione dei problemi e attività di miglioramento di processo e prodotto:**
 - le modifiche devono essere effettuate su indicazioni del Verificatore.
- **stabilire l'indipendenza del controllo della qualità dagli altri processi del ciclo di vita:**
 - il Verificatore deve essere una persona fisica diversa da quella che ha redatto il documento.

2) **eseguire valutazioni di processi, prodotti e servizi:**

- a) il Verificatore deve valutare i prodotti e i servizi al fine di garantirne la conformità rispetto ai criteri stabiliti;
- b) il Verificatore deve assicurarsi che la verifica e la convalida degli output dei processi del ciclo di vita siano eseguiti conformemente con quanto concordato in precedenza;
- c) il Verificatore deve applicare il processo di misurazione della qualità per verificare che il prodotto o servizio rispetti le metriche precedentemente stabilite;
- d) il Verificatore deve esprimere un giudizio e segnalare eventuali problematiche riscontrate.

3) **gestire report e record del Controllo della Qualità:**

- **l'attività consiste in:**
 - a) **stilare report e record relativi alle attività di Controllo della Qualità:**

- i report e i record vengono generati tramite l'utilizzo coordinato di Jira_G, Google Sheets_G e Grafana_G che porta alla creazione di un cruscotto di qualità_G. Si rimanda a Sezione 4.8.3.1 per ulteriori dettagli.
- b) **mantenere, archiviare e distribuire i report:**
 - il cruscotto di qualità_G è consultabile in ogni momento da ogni membro del gruppo, che possiede un link per accedervi. L'aggiornamento del cruscotto avviene in automatico e in tempo reale grazie al suo collegamento con Jira_G.
- c) **identificare incidenti e problemi associati alle valutazioni effettuate:**
 - in sede di retrospettiva, il gruppo lavora in modo coordinato per identificare gli incidenti e i problemi, servendosi delle informazioni presentate dal Verificatore tramite il cruscotto di qualità_G;
 - l'Amministratore si occupa della redazione di un verbale_G contenente anche gli esiti del processo di Controllo della Qualità.
- d) **trattare incidenti e problemi:**
 - a) in caso di segnalazione di incidenti e problemi deve essere svolto un lavoro collettivo per la loro risoluzione;
 - b) tutte le criticità devono prevedere risoluzioni e arginamenti già predisposti all'interno del documento *Piano di Progetto_G v1.15.0*.
 - c) nel caso tali problemi o incidenti siano di carattere generale, deve essere avvisato collettivamente il gruppo dell'insorgenza di tali problemi o incidenti, al fine di evitare future ricorrenze degli stessi.

4.8.3.1 Tecnologie

4.8.3.1.1 Jira_G

Jira_G, essendo l'ITS_G del gruppo, è la fonte principale di informazioni per il cruscotto di qualità_G.

4.8.3.1.2 Google Sheets_G

Google Sheets_G viene utilizzato per rendere meglio manipolabili i dati provenienti da Jira_G, in modo da poterli analizzare con più facilità e calcolare comodamente metriche come CPI_G, EAC_G, EV_G.

4.8.3.1.3 Grafana_G

Grafana_G è l'applicazione utilizzata per visualizzare i dati raccolti tramite l'implementazione di un cruscotto di qualità_G. Le informazioni mostrate sono le seguenti:

- sprint_G rimanenti;
- budget rimanente;
- rapporto EAC_G e BAC;
- andamento CPI_G;
- rapporto PPV_G, PAC_G e PEV_G;
- metriche soddisfatte_G;
- ore svolte e rimanenti per ruolo;
- task_G svolti;
- story_G point per stato;

- bug_G aperti.

I dati sono visualizzati sotto forma di grafici, per aiutarne l'analisi e la comprensione, e sono consultabili in ogni momento dal gruppo.

5 Processi tecnici

5.1 Processo di analisi della missione

Tailored conformance on ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.1 per ISO_G/IEC_G/IEEE 12207:2017 Annex A

5.1.1 Scopo

Il processo di analisi della missione definisce i problemi e le opportunità dai quali scaturisce il progetto, caratterizza lo spazio delle soluzioni e determina una classe di soluzione preferita.

5.1.2 Strategia di identificazione e analisi della missione

5.1.2.1 Sistemi e servizi abilitanti

Gli strumenti di comunicazione adottati dal gruppo sono descritti nella Sezione 3.1.2.3.1.

5.1.2.2 Opportunità

Si analizzano i problemi e le opportunità per acquisire una panoramica completa del contesto presentato dal Capitolo_G.

Si identifica l'ambito del Capitolo_G attraverso la definizione di:

- difetti e *pain point* delle soluzioni pre-esistenti, se disponibili;
- soluzioni alternative, con vantaggi e svantaggi rispetto alle soluzioni pre-esistenti, se disponibili;
- contesto tecnologico di applicazione;
- tipologia di utenza attesa;
- destinazione d'uso del prodotto finale.

La sintesi di bisogni e requisiti avviene nel contesto del processo di definizione di bisogni e requisiti degli Stakeholder disponibile nella Sezione 5.2.

5.1.2.3 Classi di soluzione

Si identificano classi di soluzione che possano sfruttare le opportunità e risolvere i problemi individuati.

Le classi di soluzione possono comprendere lo sviluppo o la modifica di sistemi software pre-esistenti.

Le classi di soluzione individuano le potenziali tecnologie che ci si aspetta essere necessarie. Possono includere l'identificazione di specifici sistemi o prodotti software adatti al riutilizzo.

Nel contesto del progetto didattico_G, il gruppo accetta le classi di soluzione proposte nel Capitolo_G, eventualmente utilizzandole come base per la formulazione di classi di soluzione alternative.

5.1.2.4 Valutazione delle classi di soluzione

Si valuta ogni classe di soluzione identificata sulla base di:

- fattibilità;
- costi;

- tempi necessari;
- rischi;
- interesse tecnologico e didattico;
- pertinenza.

La valutazione delle classi di soluzione può avvenire tramite:

- studio della documentazione_G dei sistemi o prodotti software identificati;
- realizzazione di esploratori tecnologici (detti *Proof of Concept*_G o PoC_G);
- consultazione di esperti, quali Proponente e Committente_G.

Sulla base dei risultati della valutazione, il gruppo individua una classe di soluzione preferita e la presenta al Proponente per la convalida.

5.1.2.5 Analisi dei requisiti

Il documento *Analisi dei Requisiti*_G raccoglie le informazioni previste da questo processo. Il documento deve ricevere approvazione esplicita da parte degli Stakeholder coinvolti.

5.2 Processo di definizione di bisogni e requisiti degli stakeholder

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.2

5.2.1 Scopo

Il processo di definizione di bisogni e requisiti degli stakeholder definisce i requisiti di un sistema che possa fornire le funzionalità di cui gli utenti e gli stakeholder necessitano.

Il processo identifica gli stakeholder coinvolti con il sistema durante l'intero suo ciclo di vita.

Identifica inoltre i loro bisogni, li analizza e li trasforma in un insieme condiviso di requisiti che:

- esprima i comportamenti attesi che il sistema dovrà avere nel suo ambiente operativo;
- cataloghi e prioritizzi ciascun requisito_G;
- riporti le fonti di ciascun requisito_G;
- funga da riferimento per la validazione dell'implementazione di ciascun requisito_G.

5.2.2 Stakeholder

Sono identificati tutti quegli stakeholder che possiedano una forte influenza sugli obiettivi, le strategie, l'operatività e le caratteristiche del prodotto.

5.2.2.1 Matrice degli stakeholder

Si classificano gli stakeholder individuati sulla base di:

- coinvolgimento nel progetto: indica l'interesse dello stakeholder nell'ambito di progetto;
- autorità sullo sviluppo: indica il potere decisionale esercitabile sull'ambito di progetto.

Entrambe le classificazioni si strutturano su tre livelli: basso, medio, alto.

5.2.2.2 Modalità di comunicazione

Per ciascuno stakeholder si identificano i canali e la frequenza della comunicazione.

5.2.3 Strategia di identificazione, analisi e trasformazione dei bisogni

Il gruppo adotta una strategia iterativa per l'identificazione, l'analisi e la trasformazione dei bisogni in requisiti. L'approccio è finalizzato alla raccolta di feedback e prevede:

- interviste e questionari;
- studio individuale di tecnologie abilitanti e documentazione_G tecnica;
- acquisizione di conoscenze tramite workshop interni e *brainstorming*;
- osservazione delle criticità delle soluzioni software preesistenti.

Le comunicazioni con gli stakeholder avvengono nelle modalità descritte in Sezione 5.2.2.2.

Le attività sono supportate, quando utile, da documenti, immagini, dimostratori tecnologici e in generale qualsiasi elemento informativo utile alla comprensione dei bisogni degli stakeholder. Le informazioni sono organizzate in modo da supportare l'identificazione, l'analisi e la trasformazione dei bisogni in requisiti. Il livello di astrazione adottato può differire in base all'interlocutore e al progresso globale conseguito dal *Processo di definizione di bisogni e requisiti degli stakeholder*.

Lo strumento adottato a supporto di queste operazioni è Miro_G.

I bisogni espressi da Committente_G e Fornitore sono raccolti, catalogati, analizzati ed espressi nel documento di *Analisi dei Requisiti*_G prodotto dal gruppo.

Alcuni stakeholder possono avere interessi avversi a quelli del gruppo o in contrasto con gli interessi di altri stakeholder. Qualora gli interessi degli stakeholder siano tra di essi contrastanti, ma non siano avversi al gruppo o al sistema software, il gruppo si adopera per mediare i contrasti. La strategia di mediazione prevede l'identificazione di un sottoinsieme di interessi e bisogni comuni, il confronto con le parti e la definizione di strategie di mediazione calate nella fattispecie.

Gli intenti o i desideri di chi si oppone al gruppo o ad uno o più dei processi di ciclo di vita del sistema software sono affrontati tramite il processo di Gestione dei Rischi disponibile alla Sezione 4.4.

Il negoziato tra le parti potrebbe essere richiesto per mediare posizioni mutualmente incompatibili, o a causa di vincoli o budget insufficiente. Anche la data di consegna prevista incide sulla realizzazione dei requisiti. Sarà sempre necessario consultare gli stakeholder coinvolti per raggiungere un accordo. Le decisioni saranno tracciate e rese disponibili agli stakeholder.

5.2.3.1 Identificazione dei bisogni

Include l'elicitazione dei bisogni direttamente dagli stakeholder o dalla documentazione_G fornita da essi, oppure la raccolta di bisogni impliciti basati sul dominio applicativo ed i contesti tecnologico, legislativo, normativo.

I bisogni degli stakeholder scaturiscono da fonti diverse. Il gruppo si impegna ad esplorare e valutare, al fine di identificare possibili bisogni, almeno questi frangenti:

- obiettivi di alto livello che il sistema dovrebbe conseguire;
- contributi concreti che il sistema dovrebbe apportare a beneficio degli stakeholder;
- scenari operativi, utili per limitare l'ambito e comprendere le aspettative e i bisogni;
- scenari operativi quotidiani, utili per assegnare una priorità ai bisogni;
- tipologie e caratteristiche degli utenti;
- ambiente operativo e contesto d'utilizzo;
- aspettative sulle prestazioni e la disponibilità del sistema;
- pratiche di business;
- norme, leggi o altri vincoli esterni.

5.2.3.2 Definizione delle priorità

Le preferenze espresse dagli stakeholder, coadiuvate dal processo di gestione delle decisioni (Sezione 4.3), guidano la selezione e la prioritizzazione dei requisiti.

5.2.3.3 Casi d'uso

Si definisce un insieme di casi d'uso (anche detti use case_G, abbreviato in UC_G) che identifichi tutte le funzionalità attese. I casi d'uso sono definiti ed utilizzati nel documento di *Analisi dei Requisiti*_G. Essi sono:

- fonte di bisogni e, indirettamente, di requisiti;
- un ausilio per l'esplorazione degli aspetti descritti nella Sezione 5.2.3.1.

Ogni caso d'uso comprende:

- 1) Codice identificativo;
- 2) Titolo;
- 3) Descrizione;
- 4) Attore_G;
- 5) Precondizioni (opzionale);
- 6) Postcondizioni (opzionale);
- 7) Scenario principale;
- 8) Scenari alternativi (opzionale);
- 9) Generalizzazioni (opzionale).

Il codice identificativo assume l'aspetto UC_G-X.Y, dove UC_G- è la radice del codice; X è una cifra positiva crescente di cifre che identifica un caso d'uso; Y è una cifra positiva crescente, attiva solo per le generalizzazioni di uno stesso caso d'uso.

I casi d'uso sono arricchiti con diagrammi realizzati secondo la sintassi Unified Modeling Language_G (UML_G) 2.0.

In nessun caso i casi d'uso propongono o presumono soluzioni implementative.

5.2.3.4 Identificazione dei vincoli

I vincoli sono un tipo di requisito_G. Possono derivare da:

- stakeholder;
- sistemi software a supporto dei processi di ciclo di vita;
- budget disponibile;
- considerazioni su prestazioni, affidabilità, sicurezza, disponibilità, manutenibilità;
- altre attività dei processi di ciclo di vita.

Sono classificati per priorità e per fonte.

5.2.3.5 Analisi dei requisiti

Il documento *Analisi dei Requisiti*_G raccoglie le informazioni previste. Il documento deve ricevere approvazione esplicita da parte degli stakeholder coinvolti.

5.3 Processo di definizione dell'architettura

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.4

5.3.1 Scopo

Il processo di definizione dell'architettura ha come scopo la creazione di uno o più modelli architeturali, nei quali si descrivono più proposte di organizzazione del prodotto dal punto di vista della comunicazione tra le sue componenti. Il modello architettuale definitivo viene deciso in collaborazione con il Proponente, in quanto deve soddisfare i suoi bisogni oltre che i requisiti individuati.

Per avere una migliore comprensione del problema da affrontare e trovare così una soluzione soddisfacente, è possibile iterare questo processo insieme ai processi di analisi della missione (Sezione 5.1), di definizione di bisogni e requisiti degli stakeholder (Sezione 5.2) e di definizione del design (Sezione 5.4).

5.3.2 Preparazione e requisiti

Il Progettista è il ruolo principale all'interno di questo processo. Il numero di persone assegnate a questo ruolo e le attività che esso dovrà svolgere vengono definite in maniera precisa durante la riunione di retrospettiva e la pianificazione dello *Sprint_G* che la segue.

È importante che il Progettista abbia una buona conoscenza dei pattern architeturali e dei concetti principali che riguardano la progettazione architettuale. In particolare, di ogni pattern deve conoscere:

- definizione;
- aspetti positivi e negativi.

È essenziale, inoltre, la conoscenza degli obiettivi del prodotto e lo studio del documento *Analisi dei Requisiti_G v2.0.1* e delle tecnologie che il gruppo ha selezionato per la sua implementazione.

5.3.3 Individuazione dell'architettura

Successivamente all'acquisizione delle conoscenze necessarie, il Progettista può procedere all'individuazione di una o più proposte architeturali per il prodotto. In questa attività, oltre alla conoscenza dei pattern descritta nella Sezione 5.3.2, sarà necessario analizzare anche gli aspetti positivi e negativi del pattern nel contesto del prodotto da sviluppare, in modo da individuare le alternative più adatte.

Le proposte architeturali individuate saranno poi illustrate al Proponente durante un meeting, il quale potrà contribuire alla decisione dell'architettura definitiva del prodotto. Il gruppo, una volta individuata l'architettura finale, dovrà descriverne i dettagli all'interno del documento *Specifica Tecnica* (Sezione 5.3.3.1).

È importante che nella scelta architettuale si mantenga un certo focus sulle caratteristiche principali di una buona architettura, riportate nell'elenco che segue:

- **sufficienza:** deve soddisfare i requisiti;
- **comprensibilità:** deve essere capita da tutti gli stakeholder;
- **modularità:** deve essere composta da parti distinte;
- **robustezza:** deve essere capace di gestire errori e condizioni limite;
- **flessibilità:** deve essere facilmente modificabile;
- **riusabilità:** le sue parti possono essere impiegate in altre applicazioni;
- **disponibilità:** la sua manutenzione causa poca indisponibilità totale;
- **sicurezza rispetto a malfunzionamenti:** deve funzionare anche in presenza di guasti;
- **semplicità:** ogni parte contiene solo il necessario e niente di superfluo;

- **incapsulazione:** deve nascondere all'esterno le parti interne dei suoi componenti;
- **coesione:** deve raggruppare le parti che svolgono gli stessi compiti;
- **basso accoppiamento:** parti distinte dipendono poco o niente le une dalle altre.

5.3.3.1 Specifica Tecnica

Insieme all'architettura del prodotto costituisce uno degli output principali del processo. In questo documento viene descritto il prodotto dal punto di vista tecnico, ovvero dal lato tecnologico e architetturale.

Gli argomenti principali trattati nel documento sono due:

- 1) **tecnologie:** vengono presentate le tecnologie utilizzate dal gruppo all'interno del prodotto, ciascuna con relativa descrizione, versione_G di utilizzo e motivazione della scelta;
- 2) **architettura:** vengono presentate:
 - a) **architettura logica:** descrizione dell'architettura del prodotto per componenti, ruoli, connessioni e interazioni;
 - b) **architettura di deployment:** descrizione della procedura di deployment.

Nel documento dovranno essere descritti nel dettaglio i design pattern utilizzati nel prodotto e derivati dalle tecnologie, inserendo anche i relativi diagrammi UML_G, e ogni altro aspetto progettuale che sia rilevante ai fini dell'architettura e del design del prodotto.

5.4 Processo di Definizione del Design

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.5

5.4.1 Scopo

Lo scopo del processo di Definizione del Design è di fornire dati e informazioni riguardo il sistema e i suoi elementi per consentirne l'implementazione coerente con le entità architettureali definite nei modelli e nelle viste dell'architettura del sistema.

5.4.2 Risultati

Come risultato dell'efficace attuazione del processo di Definizione del Design:

- vengono definite le caratteristiche di ogni elemento del sistema;
- vengono allocati i requisiti software negli elementi di sistema;
- vengono scelti i software da utilizzare per la creazione dei diagrammi a supporto del design;
- vengono definite le interfacce fra gli elementi del sistema;
- vengono valutate le alternative di design;
- vengono disegnati i diagrammi a supporto del design;
- viene effettuato il tracciamento fra i requisiti software e gli elementi di sistema.

Le scelte di design relative agli elementi di sistema e i relativi diagrammi sono illustrati nel documento *Specifica Tecnica*.

5.4.3 Attività

5.4.3.1 Pianificazione della Definizione del Design

L'attività consiste nei seguenti task_G:

1) definire la strategia della Definizione del Design:

la strategia del processo di Definizione del Design consiste in:

- decomporre il sistema in parti componibili;
- organizzare di tali componenti (stabilendo ruoli, responsabilità e interazioni);
- stabilire le interfacce fra i componenti;
- stabilire i paradigmi di composizione per collegare i componenti fra loro.

I risultati di questa progettazione sono illustrati nei rispettivi diagrammi. Viene anche valutato, se necessario e possibile, l'uso dei design pattern, illustrati nel testo *Design Patterns: Elements of Reusable Object-Oriented Software (1994)*, per standardizzare e ottimizzare l'implementazione delle interazioni tra le componenti del sistema;

2) selezionare principi e caratteristiche di design:

i Progettisti devono seguire concetti base fondamentali come l'astrazione, la modularizzazione, l'incapsulamento, la separazione tra interfaccia e implementazione. Inoltre devono rispettare i principi della SOLID programming, per creare software più modulare, scalabile, manutenibile e adattabile ai cambiamenti futuri;

3) scegliere le tecnologie a supporto della Definizione del Design:

la scelta delle tecnologie deve seguire questa prassi:

- i Progettisti devono individuare le possibili tecnologie in base alle loro conoscenze pregresse e ad un'esplorazione tecnologica, la discussione fra loro deve avvenire nei canali Discord_G "database_G" e "mvp_G";
- deve essere eseguito un breve studio di fattibilità per verificare la possibilità di utilizzo delle tecnologie;
- nei canali Discord_G "mvp_G" e "database_G", i Progettisti devono chiedere un confronto con gli altri membri del gruppo per valutare alternative tecnologiche o confermare le scelte fatte.

5.4.3.2 Progettazione degli elementi del sistema

L'attività consiste nei seguenti task_G:

1) definire gli strumenti di design necessari:

- riguardo i software, deve essere utilizzato StarUML per la modellazione di diagrammi E-R e delle classi. Figma_G deve essere impiegato per la realizzazione di prototipi di interfaccia grafica;
- riguardo i linguaggi, nei diagrammi di StarUML deve essere utilizzato l'UML_G.

2) trasformare le caratteristiche architetturali e di design nella progettazione in elementi del sistema:

- le caratteristiche architetturali e di design sono trasformate in unità architetturali, rappresentate nei relativi diagrammi.

Le unità architetturali sono unità funzionali (o di responsabilità) ben definite, realizzabili da un singolo programmatore. A una singola unità architettureale possono corrispondere uno o più moduli di codice, la cui corrispondenza unità – modulo è determinata dalle caratteristiche del linguaggio di programmazione utilizzato per la realizzazione;

3) esaminare le alternative di progettazione e la fattibilità dell'implementazione:

- i Progettisti devono identificare le alternative di design;

- le alternative di design devono essere messe in confronto fra loro per arrivare ad una soluzione finale ottimale;
- deve essere preso in considerazione anche il parere degli altri membri del gruppo in fase di retrospettiva.

4) **definire o ridefinire le interfacce fra gli elementi del sistema ed elementi esterni:**

- le interfacce devono essere identificate e definite nel processo di Definizione dell'Architettura al livello o all'estensione necessari per l'intento architetturale e la comprensione;
- queste vengono devono essere raffinate nel processo di Definizione del Design basato sulle caratteristiche di progettazione, interfacce e interazioni degli elementi software con gli altri elementi che lo compongono e con entità esterne;
- possono essere identificate e definite ulteriori interfacce che non sono state affrontate nella Definizione dell'Architettura.

5) **stabilire gli artefatti di design:**

gli artefatti di design prodotti da questo processo sono i diagrammi E-R per la modellazione del database_G, i diagrammi delle classi per la rappresentazione delle entità e delle loro relazioni nel sistema software, e dei mockup o wireframe per i prototipi dell'interfaccia grafica.

5.4.3.3 Studio e scelta delle tecnologie mirate all'implementazione

L'attività consiste nei seguenti task_G:

1) **identificazione delle tecnologie:**

le tecnologie vengono individuate fra quelle proposte dal Proponente, fra le conoscenze pregresse del gruppo e fra quelle rilevate in un primo periodo di esplorazione;

2) **studio delle tecnologie:**

- per ogni tecnologia deve essere realizzato un PoC_G, di piccola dimensione ed elevata semplicità finalizzato ad effettuare uno studio iniziale di fattibilità e ad una familiarizzazione con le nuove tecnologie. I PoC_G sono contenuti nel repository_G omonimo, nel loro relativo branch_G. Per il loro sviluppo sono impiegati anche i Programmatori;
- i PoC_G devono essere integrati in un unico prodotto software, per studiare l'integrazione fra i diversi componenti dello stack tecnologico.

3) **scelta delle tecnologie:**

la scelta finale delle tecnologie avviene in seguito a discussioni interne fra i membri del gruppo e esterne con Committente_G e Proponente, che ne mettono in luce aspetti positivi e negativi relativamente alla dimensione del progetto.

5.4.3.4 Gestione del design

L'attività consiste nei seguenti task_G:

1) **giustificare le scelte di design:**

le scelte di design devono essere giustificate e documentate nel documento *Specific Tecnica*. Le giustificazioni possono includere confronti con alternative possibili e analisi delle implicazioni a lungo termine delle decisioni prese. Questo livello di dettaglio aiuta a garantire che il design sia robusto, efficiente e in grado di soddisfare le esigenze del progetto in modo efficace;

2) **stabilire il tracciamento tra gli elementi di design, i requisiti del sistema e le entità architetture dell'architettura del sistema software;**

3) **determinare lo stato del sistema software e del design degli elementi:**

il processo di Misurazione deve essere utilizzato per stabilire misure per la completezza e la qualità del design mentre progredisce. I processi di Verifica e Validazione vengono attivati per verificare e convalidare il design dettagliato e l'implementazione. Ciò include la valutazione periodica delle caratteristiche di progettazione nel caso di evoluzione del sistema software e della sua architettura, nonché la previsione dell'eventuale obsolescenza dei componenti e delle tecnologie, la loro sostituzione nel tempo nel ciclo di vita del sistema software e le conseguenze per la definizione del design. Il processo di Gestione dei Rischi viene tipicamente applicato per valutare i rischi nella strategia di design;

4) **fornire gli artefatti e le informazioni di design:**

- il processo di Gestione della Configurazione deve essere utilizzato per stabilire e mantenere elementi di configurazione e le linee guida per artefatti come modelli di progettazione;
- il processo di Gestione delle Informazioni deve controllare gli elementi informativi, come descrizioni di progettazione e specifiche;
- gli artefatti di design devono essere resi disponibili nel documento *Specifica Tecnica*.

5.5 Processo di implementazione

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.7

5.5.1 Scopo

Lo scopo del processo di implementazione è di concretizzare un elemento specifico del sistema. Questo processo traduce i requisiti, le architetture e i design (inclusendo le interfacce), in azioni che danno vita ad un elemento di sistema in linea con le pratiche della tecnologia implementativa selezionata, facendo uso delle specializzazioni e discipline tecniche appropriate.

Il risultato di questo processo è un elemento del sistema che soddisfa i requisiti specifici e derivati del sistema, nonché le architetture e i design definiti.

5.5.2 Risultati

La corretta realizzazione del processo di implementazione produce i seguenti risultati:

- identificazione dei vincoli implementativi che influenzano i requisiti, l'architettura o il design;
- realizzazione di un elemento di sistema;
- tracciamento dell'implementazione di tale elemento.

5.5.3 Attività

5.5.3.1 Progettazione dell'elemento da implementare

È necessario che prima dell'inizio effettivo dello sviluppo di un elemento software, questo sia stato definito dai processi di definizione dell'architettura (Sezione 5.3) e del design (Sezione 5.4).

Questi processi sono fondamentali, poiché al loro completamento si dispone già di una struttura ben definita del sistema, che include le sue componenti principali, le loro interazioni e le funzionalità che devono essere implementate.

Questo approccio fornisce quindi una guida chiara per gli sviluppatori durante l'implementazione, riducendo il rischio di deviazioni o malintesi in quanto trasforma il processo di implementazione in una traduzione del design in codice.

5.5.3.2 Preparazione per l'implementazione

Fondamentale per garantire il livello di qualità richiesto dal *Piano di Qualifica_G* è definire test specifici per l'elemento di sistema che si vuole realizzare secondo il modello di sviluppo *Test Driven Development*. Questo si concretizza nella realizzazione da parte del Progettista di una serie di test di unità_G precedentemente allo sviluppo vero e proprio dell'elemento di sistema. Questo vale anche per codice riutilizzato o codice esterno che viene adattato per soddisfare i requisiti richiesti.

In caso di modifica è necessario reiterare l'esecuzione dei test sviluppati, al fine di garantire che il software sviluppato e testato in precedenza funzioni ancora come previsto. In caso contrario, si parlerebbe di regressione.

Considerando la complessità degli elementi grafici e delle interazioni utente_G che non possono essere facilmente o economicamente testati in modo automatico, viene istituita una modalità di testing_G manuale, che deve essere eseguita prima dallo sviluppatore dell'elemento di sistema (Programmatore) durante l'implementazione, e successivamente dal Verificatore, quando controlla il lavoro presente nella Pull Request_G relativa. Questo vale sia per il codice sviluppato in proprio sia per quello riutilizzato o esterno, che viene adattato per soddisfare i requisiti specifici.

Per aiutare il lavoro di verifica da parte del Verificatore, vengono riportate in Pull Request_G tutte le funzionalità che si ha necessità di controllare manualmente, oltre al riferimento al task_G interessato su Jira_G.

Dopo aver sviluppato e testato i singoli moduli attraverso i test di unità_G, inizia il processo di integrazione (Sezione 5.6).

5.5.3.3 Implementazione

Decisa una strategia di testing_G per l'elemento di sistema e scritti i relativi test di unità_G, il Programmatore può quindi procedere allo sviluppo o all'adeguamento del software.

Per garantire una standardizzazione e una migliore gestione del codice, devono essere adottate le seguenti norme e pratiche di organizzazione:

- 1) ciascuna categoria di elementi deve essere assegnata a una cartella specifica (Sezione 4.5.3.1.3.1):
 - tutti i file correlati ad uno specifico pattern devono essere organizzati all'interno di una cartella dedicata esclusivamente a quel pattern;
 - tutti i file riguardanti l'interfaccia utente_G devono essere raggruppati in una cartella apposita;
 - se all'interno di una cartella sono presenti più file legati ad una stessa classe del modello, devono essere raggruppati in una sottocartella dedicata.
- 2) ogni componente e ogni classe deve essere implementata in un proprio file. Questa pratica semplifica la gestione e la ricerca dei componenti all'interno del progetto, garantendo una maggiore chiarezza e coerenza nel codice. Può essere fatta un'eccezione a questa norma nel caso in cui vengano adottati design pattern che risultano più manutenibili se le loro classi sono implementate tutte nello stesso file;

3) i nomi di variabili, metodi e funzioni devono essere parlanti, ovvero devono riflettere il loro scopo e la loro funzione all'interno del sistema, così da facilitare la comprensione e manutenzione del codice;

4) regole di codifica:

- type safety con Typescript attraverso l'utilizzo di tipi specifici e limitando al massimo l'uso del tipo `any`;
- notazione camel per nomi di file e variabili;
- indentazione del codice di tipo Egyptian braces;
- limitare il più possibile la lunghezza di metodi e delle singole righe di codice, perseguendo i principi di leggibilità e manutenibilità. **Riferimento assente**

5) buone pratiche per la codifica:

- sviluppo di test automatici al fine di verificare il comportamento delle singole unità di codice (Sezione 5.7);
- identificazione e rimozione del codice duplicato per ridurre la complessità e migliorare la manutenibilità. Questo può essere fatto estraendo il codice duplicato in funzioni o classi riutilizzabili.

Prima di sottoporre il software a verifica è necessario assicurarsi che l'elemento di sistema non regredisca le funzionalità tramite l'esecuzione di `testingG` dedicato.

5.5.3.4 Gestione dei risultati dell'implementazione e delle anomalie incontrate

L'elemento implementato deve essere approvato durante il processo di verifica (Sezione 5.7) e integrato nel sistema come descritto nella Sezione 5.6.

5.6 Processo di integrazione

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.8

5.6.1 Scopo

Il processo di integrazione ha lo scopo di combinare iterativamente un insieme di elementi software al fine di ottenere un prodotto che soddisfi i requisiti, rispettando l'architettura e il design definiti.

Di conseguenza, esso viene attuato in coordinamento con:

- il processo di definizione di bisogni e requisiti degli stakeholder (Sezione 5.2);
- il processo di definizione dell'architettura (Sezione 5.3).
- il processo di definizione del design (Sezione 5.4);

L'integrazione del sistema software avviene automaticamente mediante strumenti che permettano la Continuous Integration_G.

5.6.2 Risultati

Come risultato della corretta implementazione del processo di integrazione:

- i nuovi elementi software implementati sono integrati con la *codebase* esistente;
- il sistema software è composto dall'integrazione degli elementi software implementati;
- i test atti a garantire il corretto funzionamento del prodotto a seguito dell'integrazione sono eseguiti automaticamente;

- le interfacce tra gli elementi software sono definite, implementate e aggiornate;
- le interfacce tra il sistema software e gli elementi esterni sono definite, implementate e aggiornate;
- gli strumenti di supporto al processo di integrazione sono individuati, configurati e in funzione;
- i risultati dell'integrazione sono identificati, assieme ad eventuali anomalie;
- i risultati dei test sono registrati e possono essere visualizzati dai membri del gruppo.

5.6.3 Attività

5.6.3.1 Definizione della strategia di integrazione

La strategia individuata deve garantire il coordinamento tempestivo delle attività dei processi di implementazione e di integrazione.

La strategia di integrazione definita dal gruppo si basa sull'adozione delle pratiche di *Continuous Integration_G* (CI) e *Continuous Deployment* (CD), al fine di permettere un frequente allineamento degli ambienti di lavoro tra i membri e garantire che la versione_G più aggiornata del prodotto sia sempre disponibile, funzionante e verificata.

Tale strategia prevede l'esecuzione di test di unità_G e di integrazione negli ambienti locali di sviluppo e nel repository_G centrale, tramite GitHub Actions_G.

La strategia di integrazione prevede il coordinamento con:

- il processo di Verifica (Sezione 5.7), per:
 - 1) fornire evidenza obiettiva che il sistema software integrato soddisfi i requisiti;
 - 2) identificare e risolvere le anomalie riscontrate, sia nel software che negli elementi ad esso associati quali requisiti, test, architettura e design.
- il processo di Validazione (Sezione 5.9), per confermare che il sistema software integrato soddisfi gli scopi e le funzionalità previste;
- il processo di Controllo della Qualità (Sezione 4.8) per supportare la verifica e la validazione del sistema software integrato, individuare e risolvere i problemi e le inconformità.

5.6.3.1.1 Identificazione dei criteri di integrazione

L'integrazione degli elementi software deve avvenire in modo incrementale, e deve risultare nella creazione di un sistema software integrato eseguibile senza errori e consistente con l'architettura e il design definiti. Il soddisfacimento dei requisiti funzionali e non funzionali descritti nel documento *Analisi dei Requisiti_G v2.0.1* è conseguito incrementalmente come risultato dei passaggi di integrazione.

5.6.3.1.2 Identificazione dei servizi abilitanti

Il gruppo si dota dei seguenti strumenti e servizi abilitanti per implementare la strategia di integrazione:

- GitHub Actions_G, per l'automazione delle attività di integrazione e di verifica nel repository_G GitHub_G;
- Node.js_G (npm) e Jest_G, per l'esecuzione dei test negli ambienti di sviluppo locali;
- Coveralls, per la valutazione della copertura dei test eseguiti;
- Jira_G, per la registrazione dei risultati dei test e delle anomalie riscontrate.

L'uso degli strumenti e dei servizi abilitanti è regolato nell'ambito del processo di verifica (Sezione 5.7).

5.6.3.2 Attuazione della strategia di integrazione

L'integrazione degli elementi software implementati avviene attuando quanto prescritto dalla strategia di integrazione, utilizzando le procedure e i servizi abilitanti individuati.

L'attuazione successiva della strategia di integrazione prosegue fino a che il sistema software completo risponde a tutte le caratteristiche individuate e descritte nel documento *Analisi dei Requisiti_G v2.0.1*.

L'integrazione degli elementi software, individuati e specificati tramite i processi di definizione dell'architettura (Sezione 5.3) e di definizione del design (Sezione 5.4), avviene elemento per elemento, secondo l'ordine di implementazione stabilito durante la pianificazione delle attività. Qualora un elemento software non fosse ancora disponibile per l'integrazione, la funzionalità può essere temporaneamente simulata tramite l'uso di *mock* o *stub*. Prima di convalidarne la conformità, il sistema software così integrato viene sottoposto a verifica e validazione per garantire che soddisfi i requisiti e le aspettative degli stakeholder.

A seguito di tali attività, il Programmatore invia i cambiamenti proposti al repository_G, tramite il meccanismo descritto nella Sezione 4.5.3.3. Il Verificatore può approvare la Pull Request_G e, tramite la funzione di *merge*, integrare l'elemento software implementato nel sistema software. A supporto di tale operazione, è prevista l'esecuzione di GitHub Actions_G che provvedono a:

- creare e associare al sistema integrato, tramite la action `tag_semver.yml`, un tag di versione_G semantica per identificare univocamente la release. Il versionamento_G avviene in accordo con la Sezione 4.5.3.1.1.2. Lo step di versionamento_G predefinito è di livello "patch" (es. `1.0.0` → `1.0.1`), ma può essere esplicitato dal Programmatore nel messaggio di commit_G tramite i modificatori:
 - `#patch`, per incrementare il numero di patch (*z*);
 - `#minor`, per incrementare il numero di minor (*y*).
- costruire un artefatto sotto forma di immagine Docker_G, e pubblicarlo sui *container registries* Docker_G Hub e GitHub_G Container Registry tramite la action `build_docker` (`build_docker.yml`);
- calcolare la copertura dei test eseguiti sul sistema integrato con Coveralls, e pubblicarla come report sulla Pull Request_G tramite la action `test_nodejs.yml`;

5.6.3.3 Gestione dei risultati di integrazione

I risultati delle attività di integrazione includono:

- il sistema software integrato;
- il responso dell'esecuzione delle attività di testing_G automatico;
- il risultato delle attività di verifica e validazione svolte dal Verificatore;
- l'identificazione e la risoluzione dei problemi di integrazione tra elementi software;
- quando applicabile, l'identificazione e la risoluzione delle anomalie imputabili agli strumenti o alle modalità di integrazione adottate (si rimanda al processo di Controllo della Qualità, Sezione 4.8).

Il sistema software integrato rappresenta una **baseline_G di prodotto**.

I risultati sono registrati nel repository_G GitHub_G e nell'ITS_G adottato dal gruppo. Quando necessario, i risultati delle attività di integrazione possono essere condivisi con:

- l'Amministratore per la risoluzione di problemi legati agli strumenti abilitanti o per l'adozione di altre misure correttive, in accordo con le parti coinvolte;
- il Proponente e/o il Committente_G per la convalida del sistema software integrato.

5.6.3.3.1 Tracciabilità degli elementi del sistema software

La tracciabilità tra gli elementi del sistema software, la Pull Request_G associata, e i requisiti avviene per mezzo di Jira_G. I ticket_G direttamente associabili ad un requisito_G ne riportano il codice identificativo nell'apposito campo. In questo modo, è possibile risalire a quali elementi del sistema software soddisfino un determinato requisito_G e, di conseguenza, quale sia lo stato di avanzamento del progetto in termini di requisiti soddisfatti.

Tramite GitHub_G, è inoltre reso disponibile un registro delle Pull Request_G approvate (e dunque integrate) o chiuse/rifutate, liberamente consultabile da tutti i membri del gruppo.

Dopo l'approvazione, a ciascun sistema software risultato dell'integrazione di elementi software implementati sarà associato un numero di versione_G (come descritto nella Sezione 4.5.3.1.1.2). Questo numero di versione_G sarà utilizzato per identificare il sistema software e per tracciare le modifiche apportate nel tempo. In accordo con quanto descritto in questa sezione, ad un numero di versione_G superiore corrisponderà un numero di requisiti soddisfatti superiore.

Ciascuna versione_G del sistema software integrato, intesa come artefatto eseguibile, dovrà essere conservata e resa disponibile per il download tramite le apposite funzionalità di GitHub_G.

I risultati del processo di integrazione vengono visualizzati su GitHub_G come resoconto delle automazioni eseguite a seguito dell'apertura o chiusura di una Pull Request_G. Le GitHub Actions_G devono prevedere la visualizzazione di messaggi che descrivano gli eventuali errori insorti oppure, in loro assenza, la corretta esecuzione dell'integrazione.

5.7 Processo di verifica

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.9

5.7.1 Scopo

Il processo di verifica ha lo scopo di dimostrare, con evidenza obiettiva, che il sistema, o una sua parte, adempie ai requisiti associati definiti nel documento *Analisi dei Requisiti_G v2.0.1*, nel rispetto delle metriche di qualità del prodotto definite nel *Piano di Qualifica_G v1.5.0*. Nel caso in cui vengano identificati errori o mancanze, vengono fornite le informazioni necessarie a determinare la risoluzione delle anomalie riscontrate.

Tali operazioni sono possibili mediante test automatici eseguiti, in parte, durante il processo di integrazione (Sezione 5.6), e test manuali eseguiti dai componenti del gruppo, con l'obiettivo di individuare tempestivamente eventuali problemi.

5.7.2 Risultati

Come risultato della corretta applicazione del processo di verifica, per ogni elemento:

- viene effettuata una verifica manuale da parte di uno o più Verificatori;
- vengono eseguiti i test automatici di unità e di integrazione;
- vengono rilevati eventuali problemi legati al rispetto della correttezza, dei requisiti o dell'architettura, in funzione di quanto definito;
- in caso di errori, vengono riportati i dati che forniscono informazioni per le azioni correttive;

- in caso di assenza di anomalie, viene fornita evidenza che il sistema verificato sia conforme ai requisiti specificati.

5.7.3 Attività

5.7.3.1 Esecuzione

5.7.3.1.1 Verifiche preliminari

La prima attività di verifica avviene durante l'implementazione del software. In questa attività devono essere sviluppati, dove possibile, i test di unità_G su cui si baserà l'elemento software da implementare, sia esso un pattern o una classe, seguendo così il principio del *Test Driven Development*. I test qui sviluppati rientrano nei test automatici adottati da *Error_418*, e devono essere inseriti all'interno di una cartella dedicata nella repository_G WMS3 (Sezione 4.5.3.1.3.1), denominata `__test__`.

All'interno della cartella `__test__`, dovranno essere implementati anche i test di integrazione_G, i quali dovranno verificare la corretta interazione tra tutte le parti di sistema. Questo tipo di test dovrà essere implementato, dove necessario, con l'utilizzo di mock.

Tutti i test automatici dovranno essere implementati utilizzando il framework di testing_G Jest_G, e dovranno essere eseguibili tramite il comando

```
npm test  
  
oppure  
  
npm run test
```

Come descritto nel processo di implementazione (Sezione 5.5), talvolta non è possibile implementare test automatici poiché risulterebbe troppo complicato e oneroso. Questo avviene quando il soggetto dell'implementazione è:

- una componente grafica appartenente alla UI o all'ambiente 3D;
- una determinata interazione tra utente_G e sistema che risulta difficile da automatizzare.

In questi casi il Programmatore dovrà testare manualmente l'elemento di sistema implementato, per verificare che venga rappresentato correttamente, o che le interazioni con l'utente_G non presentino problemi.

Qualora l'impegno, temporale o economico, per la conduzione dei test manuali si rivelasse eccessivo rispetto agli impegni di progetto, *Error_418* valuterà l'introduzione di test E2E_G per la verifica.

5.7.3.1.2 Verifiche in Pull Request_G

Tutti i test di unità_G e di integrazione implementati dovranno essere eseguiti automaticamente dalle GitHub Actions_G (Sezione 4.5.3.1.3.2) ogni volta che una Pull Request_G (Sezione 4.5.3.3) viene aperta o aggiornata nel suo contenuto. L'automazione designata all'esecuzione dei test dovrà essere codificata nel file `test_nodejs.yml`, situato nella repository_G WMS3 al percorso `WMS3/.github/workflows/`. Il merging di una Pull Request_G non potrà avvenire se un test automatico fallisce.

Successivamente all'esecuzione dei test automatici, il Verificatore potrà controllare il contenuto della Pull Request_G. In particolare dovrà condurre delle verifiche manuali atte ad accertare che:

- il codice scritto rispetti quanto definito nel processo di implementazione (Sezione 5.5);
- il codice esegua senza problemi;

- l'elemento implementato nella Pull Request_G sia visualizzato correttamente all'interno del prodotto, e assolva tutte le funzionalità ad esso richieste dalla *Specifica Tecnica v1.3.0* e dall'*Analisi dei Requisiti_G v2.0.1*.

5.7.3.2 Gestione dei risultati

I risultati ottenuti dai test sono visualizzati su GitHub_G:

- all'interno della Pull Request_G, sotto forma di commenti del Verificatore, per i test manuali;
- all'interno della Pull Request_G, sotto forma di commento, oppure nella sezione relativa alle automazioni eseguite nella Pull Request_G, per i test automatici.

Se a seguito di tutti i test non emergono anomalie o errori, la Pull Request_G potrà essere accettata e chiusa dal Verificatore, il quale approverà il merge all'interno del branch_G **dev**. Al momento della chiusura dovranno essere eseguite altre automazioni riguardanti il versionamento_G (Sezione 4.5.3.1.1.2). È possibile visualizzare l'insieme delle Pull Request_G approvate ed integrate mediante merge all'interno di GitHub_G.

Nel caso in cui almeno un'attività di verifica faccia emergere problematiche, le modifiche non possono essere integrate nel branch_G principale. La segnalazione e gestione degli errori dovrà avvenire nel seguente modo:

- **errori nei test automatici:** vengono segnalati dalle GitHub Actions_G. È possibile visualizzare il log dell'esecuzione dell'automazione nella sezione Actions della repository_G. Il Programmatore dovrà quindi controllare il log ed effettuare le correzioni necessarie a risolvere l'errore che si è presentato;
- **errori nei test manuali:** vengono segnalati dal Verificatore tramite commenti nella Pull Request_G. Il Verificatore dovrà inserire dei commenti nei quali dovrà descrivere nel modo più dettagliato possibile gli errori riscontrati e i passaggi per riprodurli, evitando così ambiguità e incomprensioni. Il commento dovrà essere inserito nel file relativo all'elemento o alla funzionalità che presenta errori, e, se possibile, nella riga dove si è individuato l'errore.

Il Programmatore dovrà quindi individuare la causa di questi errori e risolverli nel minor tempo possibile, così da evitare situazioni di stallo nello sviluppo del software.

Programmatore e Verificatore si impegnano a tenere traccia di tutti i bug_G riscontrati tramite le apposite funzioni fornite dal ITS_G. Per ciascun bug_G, si richiede il tracciamento di:

- una breve descrizione del problema;
- una classificazione di urgenza e gravità, su una scala da 1 a 3;
- l'elencazione dei passaggi da svolgere per riprodurre il comportamento, quando applicabile.

La lista dei bug_G potrà essere fornita al Proponente su sua esplicita richiesta.

5.8 Processo di Transizione

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.10

5.8.1 Scopo

Il processo di Transizione ha lo scopo di stabilire la capacità del sistema di fornire i servizi specificati dalle richieste del Proponente (descritti in *Analisi dei Requisiti_G v2.0.1*), all'interno di un ambiente operativo.

Coinvolge l'installazione del sistema verificato al fine di consentire l'operatività in ambienti diversi, come quelli di sviluppo, test o rilascio in una diversa infrastruttura.

5.8.2 Risultati

I risultati della corretta applicazione del processo di transizione sono:

- avviene il rilascio di una versione_G del sistema;
- viene effettuato il deploy del software su Azure_G;
- viene garantito il funzionamento dei servizi e l'operatività del sistema;
- vengono identificati i risultati e le anomalie della transizione;
- viene stabilita la tracciabilità degli elementi trasferiti;
- viene redatto del documento *Manuale Utente_G*;
- viene fornito supporto e formazione degli utenti all'utilizzo del sistema.

5.8.3 Attività

5.8.3.1 Rilascio dell'ultima versione_G di sistema

Il repository_G GitHub_G dedicato a WMS3 è provvisto di un branch_G principale **dev**, il quale rappresenta la fonte da cui vengono creati i feature_G branch_G. Questi ultimi sono necessari al conseguimento degli obiettivi di un task_G e oggetto dei merge a seguito delle successive pull request_G.

I rilasci del prodotto devono avvenire sempre dal branch_G di partenza **dev**, in quanto il codice contenuto in esso garantisce il soddisfacimento dei requisiti di qualità imposti dal gruppo.

Ogni release avvenuta con successo, in conformità con il superamento dei test e le procedure descritte nei processi di implementazione (Sezione 5.5), integrazione (Sezione 5.6) e verifica (Sezione 5.7), deve generare una release specifica tracciata e consultabile (Sezione 5.8.3.4) con relativo numero di versione_G aggiornato (Sezione 4.5.3.1).

Ad ogni modifica del sistema devono venire aggiornati anche i documenti *Manuale Utente_G* e *Specifiche Tecnica*, al fine di includere la documentazione_G relativa alle modifiche apportate.

5.8.3.2 Deploy su Azure_G

Il gruppo ha definito una GitHub_G Action chiamata *Deploy to VPS_G* (i cui job sono descritti nel file *deploy.yml*) la quale effettua il deploy del sistema, aggiornato all'ultima versione_G rilasciata sul branch_G **dev**, all'interno della macchina virtuale di Azure_G.

Questo permette la fruizione del sistema senza la necessità di dover scaricare le ultime modifiche sul proprio dispositivo e senza dover ricreare le relative immagini Docker_G.

5.8.3.3 Redazione del documento *Manuale Utente_G*

Al fine di fornire agli utenti di WMS3 una guida documentale atta ad indicare il suo corretto utilizzo, il gruppo deve mettere a disposizione il documento *Manuale Utente_G*.

In esso deve venire riportato:

- guida all'avvio e configurazione dell'ambiente;
- descrizione delle funzionalità offerte dal sistema;
- guida dettagliata all'utilizzo delle funzionalità offerte.

Il *Manuale Utente_G* deve venire redatto seguendo i principi di redazione dei documenti descritti nel paragrafo riguardante il processo di gestione delle informazioni (Sezione 4.6).

5.8.3.4 Tracciabilità risultati

I risultati relativi a controlli e test automatici avvenuti in fase di release sono visualizzati su GitHub_G nella sezione relativa alle automazioni avvenute nella pull request_G dell'elemento in oggetto.

Se da essi non emergono anomalie o errori, la transizione è avvenuta correttamente e viene confermata la conformità degli elementi analizzati.

Per avere traccia degli elementi verificati che costituiscono le singole release del prodotto, su GitHub_G è possibile visualizzare l'insieme delle release e lo storico delle corrispondenti pull request_G approvate ed integrate.

5.9 Processo di Validazione

Conformant to outcomes to ISO_G/IEC_G/IEEE 12207:2017 clause 6.4.11

5.9.1 Scopo

Il processo di Validazione si occupa di fornire prove oggettive che il sistema soddisfi i requisiti del Proponente, descritti nel documento *Analisi dei Requisiti_G v2.0.1*.

Lo scopo del processo è quindi accertare la capacità del sistema, o di un suo elemento, di soddisfare, in specifiche condizioni operative, l'obiettivo per cui è stato creato.

5.9.2 Risultati

Come risultato della corretta implementazione del processo di Validazione si ottiene:

- individuazione di limiti e vincoli che influenzano i requisiti, il design e l'architettura del sistema;
- feedback da parte del Proponente in merito allo stato di avanzamento e qualità del lavoro svolto;
- evidenza che il sistema, o l'elemento di sistema, soddisfi i requisiti dichiarati;
- tracciamento di eventuali anomalie riscontrate.

5.9.3 Attività

5.9.3.1 Validazione interna

Dopo aver apportato avanzamenti al software e a seguito dell'attuazione del processo di verifica (Sezione 5.7), devono avvenire delle analisi e test manuali per constatare lo stato di soddisfacimento dei requisiti. I risultati di tali analisi devono essere discussi durante i meeting di retrospettiva in quanto fondamentali per la pianificazione delle future attività. I test manuali precedentemente menzionati devono avvenire mediante l'utilizzo di Docker Compose_G, permettendo così l'esecuzione su un ambiente con caratteristiche standard, e prevedono l'esecuzione manuale delle funzionalità implementate atte a soddisfare i requisiti definiti.

5.9.3.2 Validazione esterna

Al fine di esaminare lo stato di avanzamento dei lavori, il gruppo deve pianificare dei meeting settimanali con il Proponente, al quale deve venire presentato il sistema in funzione.

Il feedback ricevuto dal Proponente durante questi meeting è essenziale per il proseguimento dei lavori e la loro pianificazione. Al termine del meeting, il gruppo dovrà effettuare una riunione interna per discutere dei feedback ricevuti e valutare, se necessario, azioni correttive da applicare.

Nel momento in cui il gruppo avrà soddisfatto tutti i requisiti obbligatori definiti nel documento *Analisi dei Requisiti_G v2.0.1* e implementato tutte le funzionalità desiderate, dovrà avvenire il meeting esterno di validazione finale del software. Il gruppo presenterà il prodotto sviluppato al Proponente, il quale, a seguito di un'attenta analisi, potrà affermare il totale adempimento dei requisiti obbligatori e delle funzionalità concordate, riconoscendo il prodotto sviluppato come Minimum Viable Product _G (MVP_G) atteso dal capitolato_G.

5.9.3.3 Gestione dei risultati della validazione

Nel caso in cui il software venga riconosciuto come MVP_G e approvato definitivamente dal Proponente, il lavoro su di esso deve subire una riduzione di priorità o arrestarsi totalmente, dando maggiore importanza all'eventuale documentazione_G incompleta in vista della revisione_G PB_G. L'adempimento dei requisiti da parte del prodotto sviluppato deve essere tracciato all'interno del documento *Specifiche Tecnica v1.3.0*.

Contrariamente, nel caso in cui emergano problematiche o mancanze, esse devono venire discusse e affrontate in un meeting interno al fine di eseguire una mirata pianificazione dei task_G correttivi. Questi ultimi vengono registrati su Jira_G in modo tale da ottenere maggiore chiarezza riguardo l'organizzazione del gruppo e l'avanzamento dei lavori. Tali considerazioni e decisioni prese devono essere presenti nel verbale_G conseguente al relativo meeting interno.

6 Tracciamento paragrafi ISO_G/IEC_G/IEEE 12207:2017

La tabella di seguito riportata consente di associare ogni capitolo del documento al rispettivo capitolo dello standard di riferimento. Viene riportato anche il grado di conformità:

- **To outcomes** indica che il gruppo ha dovuto adattare lo standard al progetto, omettendo o reinterprestando sezioni incompatibili con la natura del progetto pur cercando il più possibile di perseguire l'obiettivo di qualità che lo standard impone;
- **Full** indica che il capitolo riporta fedelmente le indicazioni dello standard con poche o nessuna azione di adeguamento.

Capitolo Norme	Capitolo Standard	Conformance Level
Sezione 2.1	6.1.2 - Supply process	To outcomes
Sezione 3.1.1	6.2.1 - Life cycle model management process	To outcomes
Sezione 3.1.3	6.2.4 - Human Resource Management process	To outcomes
Sezione 3.1.4	6.2.5 - Quality Management process	To outcomes
Sezione 4.1	6.3.1 - Project Planning process	To outcomes
Sezione 4.2	6.3.2 - Project assessment and control process	Full
Sezione 4.3	6.3.3 - Decision Management process	Full
Sezione 4.4	6.3.4 - Risk Management process	Full
Sezione 4.5	6.3.5 - Configuration Management process	To outcomes
Sezione 4.6	6.3.6 - Information Management process	To outcomes
Sezione 4.7	6.3.7 - Measurement process	To outcomes
Sezione 4.8	6.3.8 - Quality Assurance process	Full
Sezione 5.1	6.4.1 - Business or Mission Analysis process	Full
Sezione 5.2	6.4.2 - Stakeholder Needs and Requirements Definition process	To outcomes
Sezione 5.3	6.4.4 - Architecture Definition process	To outcomes
Sezione 5.4	6.4.5 - Design Definition process	To outcomes
Sezione 5.5	6.4.7 - Implementation process	To outcomes
Sezione 5.6	6.4.8 - Integration process	To outcomes
Sezione 5.7	6.4.9 - Verification process	To outcomes
Sezione 5.8	6.4.10 - Transition process	To outcomes
Sezione 5.9	6.4.11 - Validation process	To outcomes

Tabella 4: Tracciamento paragrafi ISO_G/IEC_G/IEEE 12207:2017