



Norme di progetto

Gruppo N.O.S - Progetto *Emporio\lambda*

nos.unipd@gmail.com

Informazioni sul documento

Versione	3.0
Approvatori	<i>Varotto Davide</i>
Redattori	<i>Martini Matteo</i>
Verificatori	<i>Terrani Giulia</i>
Uso	Interno
Distribuzione	Gruppo N.O.S <i>Prof. Vardanega Tullio</i> <i>Prof. Cardin Riccardo</i>

Descrizione

Questo documento contiene le linee guida per la gestione di tutti i processi istanziati dal gruppo.

Registro modifiche documento

Versione	Data	Nominativo	Ruolo	Verificatore	Descrizione
3.0	2021_04_29	Varotto Davide	Responsabile	-	Approvazione del documento.
2.1	2021_03_22	Martini Matteo	Amministratore	Terrani Giulia	Corretto il documento post RP, in particolare §A.1 e §A.2. Aggiornata §2.2.5.
2.0	2021_03_06	Tredese Leonardo	Responsabile	-	Approvazione del documento.
1.4	2021_02_06	Varotto Davide	Amministratore	Terrani Giulia	Aggiornata §2.2.5 e redatta §3.3.7.2.
1.3	2021_02_04	Varotto Davide	Amministratore	Tredese Leonardo	Modificate §4.4.3, §4.2.4.3 e §4.2.4.5.
1.2	2021_02_03	Terrani Giulia	Amministratore	Brescanzin Lorenzo	Modificate §2.3.4.3, §2.3.5.3, §2.3.6.3, §3.2.3 e §3.4.5. Redatta §2.3.4.6.
1.1	2021_02_02	Martini Matteo	Amministratore	Fantinato Filippo	Redatte §2.3.4.9, §2.3.5.7, §2.3.6.6, §3.2.10, §4.2.4.7, §4.4.5. Modifica §3.3.6 e §B.2.
1.0.0	2021_01_10	Martini Matteo	Responsabile	-	Approvazione del documento.
0.2.0	2020_12_30	-	-	Panighel Cristiano	Verifica documento.
0.1.4	2020_12_24	Terrani Giulia	Amministratore	Panighel Cristiano	Redatta §B.2.
0.1.3	2020_12_24	Terrani Giulia	Amministratore	Panighel Cristiano	Redatta §A.2 e §B.1.
0.1.2	2020_12_24	Terrani Giulia	Amministratore	Martini Matteo	Inserite immagini §2.3.5.6.
0.1.1	2020_12_22	Terrani Giulia	Amministratore	Panighel Cristiano	Aggiornata §1.4.2 e §3.7.4, redazione §2.2.7, §2.3.7, §3.2.11, §3.7.5.
0.1.0	2020_12_21	-	-	Martini Matteo	Verifica §4, ultimata verifica documento.
0.0.15	2020_12_19	-	-	Tredese Leonardo	Verifica complessiva §2.
0.0.14	2020_12_17	-	-	Brescanzin Lorenzo	Verifica complessiva §1 e §3.
0.0.13	2020_12_16	Terrani Giulia	Amministratore	Panighel Cristiano	Modifica §3.5.4.1, §3.5.4.2, §3.5.5.2.

Versione	Data	Nominativo	Ruolo	Verificatore	Descrizione
0.0.12	2020_12_15	<i>Terrani Giulia</i>	Amministratore	<i>Panighel Cristiano</i>	Redazione da §3.5.1 a §3.5.5.2, §3.6, §3.7, §4.4.4, eliminate "Istituzione del processo", "Valutazione del processo" e "Miglioramento del processo".
0.0.11	2020_12_13	<i>Terrani Giulia</i>	Amministratore	<i>Fantinato Filippo</i>	Risolto problemi di impaginazione, modifica §2.2.6, redazione da §3.4.1 a §3.4.7.
0.0.10	2020_12_11	<i>Terrani Giulia</i>	Amministratore	<i>Fantinato Filippo</i>	Redazione §3.3.1 a §3.3.7.3.
0.0.9	2020_12_09	<i>Terrani Giulia</i>	Amministratore	<i>Varotto Davide</i>	Redazione §2.3.5.6 a §2.3.6.5.
0.0.8	2020_12_07	<i>Terrani Giulia</i>	Amministratore	<i>Varotto Davide</i>	Redazione §2.3.5.5, da §4.2.5.1 a §4.2.5.6.
0.0.7	2020_12_05	<i>Panighel Cristiano</i>	Amministratore	<i>Varotto Davide</i>	Redazione da §4.2.1 a §4.2.4.6, da §4.3.1 a §4.5.4.
0.0.6	2020_12_03	<i>Tredese Leonardo</i>	Amministratore	<i>Fantinato Filippo</i>	Redazione da §3.2.5 a §3.2.11.
0.0.5	2020_11_30	<i>Terrani Giulia</i>	Amministratore	<i>Varotto Davide</i>	Redazione §4.1, §3.1 a §3.2.4.
0.0.4	2020_11_29	<i>Terrani Giulia</i>	Amministratore	<i>Tredese Leonardo</i>	Redazione da §2.3.5.1 a §2.3.5.4.
0.0.3	2020_11_28	<i>Brescanzin Lorenzo</i>	Amministratore	<i>Varotto Davide</i>	Redazione da §2.3.1 a §2.3.4.8.
0.0.2	2020_11_27	<i>Fantinato Filippo</i>	Amministratore	<i>Varotto Davide</i>	Redazione §2.1 a §2.2.6.
0.0.1	2020_11_26	<i>Terrani Giulia</i>	Amministratore	<i>Panighel Cristiano</i>	Creazione documento e redazione §1.1, §1.3, §1.4.1.

Indice

1	Introduzione	6
1.1	Scopo del documento	6
1.2	Scopo generale del prodotto	6
1.3	Glossario	6
1.4	Riferimenti	6
1.4.1	Normativi	6
1.4.2	Informativi	6
2	Processi primari	8
2.1	Descrizione	8
2.2	Processo di fornitura	8
2.2.1	Scopo del processo	8
2.2.2	Descrizione della sezione	8
2.2.3	Aspettative	8
2.2.4	Rapporti con l'azienda proponente RedBabel	8
2.2.5	Materiale fornito	9
2.2.6	Studio di fattibilità	9
2.2.7	Strumenti	9
2.3	Processo di sviluppo	10
2.3.1	Scopo del processo	10
2.3.2	Descrizione della sezione	10
2.3.3	Aspettative	10
2.3.4	Analisi dei requisiti	10
2.3.4.1	Scopo dell'attività	10
2.3.4.2	Descrizione della sezione	10
2.3.4.3	Aspettative	11
2.3.4.4	Struttura del documento	11
2.3.4.5	Classificazione dei requisiti	11
2.3.4.6	Requisiti di processo	12
2.3.4.7	Classificazione dei casi d'uso	12
2.3.4.8	Qualità dell'analisi dei requisiti	13
2.3.4.9	Metriche	13
2.3.5	Progettazione	13
2.3.5.1	Scopo dell'attività	13
2.3.5.2	Descrizione della sezione	14
2.3.5.3	Aspettative	14
2.3.5.4	Qualità dell'attività di progettazione	14
2.3.5.5	Periodi della progettazione	15
2.3.5.6	Diagrammi UML 2.0	15
2.3.5.7	Metriche	21
2.3.6	Codifica software	21
2.3.6.1	Scopo dell'attività	21
2.3.6.2	Descrizione della sezione	22
2.3.6.3	Aspettative	22
2.3.6.4	Qualità della codifica	22
2.3.6.5	Convenzioni generiche	22
2.3.6.6	Metriche	22
2.3.7	Strumenti	23

3	Processi di supporto	24
3.1	Descrizione	24
3.2	Documentazione	24
3.2.1	Scopo dell'attività	24
3.2.2	Descrizione della sezione	24
3.2.3	Aspettative	24
3.2.4	Ciclo di vita di un documento	25
3.2.5	Classificazione dei documenti	25
3.2.6	Nomenclatura dei documenti	25
3.2.7	Directory dei documenti	26
3.2.8	Struttura dei documenti	26
3.2.9	Formattazione e norme tipografiche	27
3.2.9.1	Template	27
3.2.9.2	Termini di glossario	27
3.2.9.3	Stile del testo	27
3.2.9.4	Formattazione di date e orari	27
3.2.9.5	Formattazione elementi grafici	28
3.2.10	Metriche	28
3.2.11	Strumenti	28
3.3	Gestione della configurazione	29
3.3.1	Scopo dell'attività	29
3.3.2	Descrizione della sezione	29
3.3.3	Aspettative	29
3.3.4	Identificazione di configurazione	29
3.3.5	Controllo di baseline	29
3.3.6	Controllo di versione	29
3.3.6.1	Codice di versione per documenti e software	29
3.3.7	Struttura della repository	29
3.3.7.1	Repository per la documentazione	30
3.3.7.2	Repository per il Poc	30
3.3.7.3	Modifiche al repository	30
3.4	Gestione della qualità	31
3.4.1	Scopo dell'attività	31
3.4.2	Descrizione della sezione	31
3.4.3	Aspettative	31
3.4.4	Istanziamento di un processo	31
3.4.5	Svolgimento di un processo	31
3.4.6	Classificazione degli obiettivi	31
3.4.7	Classificazione delle metriche	32
3.5	Verifica	33
3.5.1	Scopo dell'attività	33
3.5.2	Descrizione della sezione	33
3.5.3	Aspettative	33
3.5.4	Analisi statica	33
3.5.4.1	Verifica della documentazione	33
3.5.4.2	Verifica del codice	33
3.5.5	Analisi dinamica	34
3.5.5.1	Tipologia di test effettuati	34
3.5.5.2	Procedura di verifica	34
3.6	Validazione	35
3.6.1	Scopo dell'attività	35
3.6.2	Descrizione della sezione	35
3.6.3	Aspettative	35
3.6.4	Validazione dei documenti	35
3.6.5	Validazione del codice	35
3.6.6	Piano di qualifica e test	35

3.6.7	Nomenclatura dei test	36
3.7	Gestione dei cambiamenti	37
3.7.1	Scopo dell'attività	37
3.7.2	Descrizione della sezione	37
3.7.3	Aspettative	37
3.7.4	Norme generali	37
3.7.5	Denominazione dei cambiamenti	37
4	Processi organizzativi	38
4.1	Descrizione	38
4.2	Gestione di processo	38
4.2.1	Scopo del processo	38
4.2.2	Descrizione della sezione	38
4.2.3	Aspettative	38
4.2.4	Pianificazione	38
4.2.4.1	Scopo dell'attività	38
4.2.4.2	Descrizione della sezione	38
4.2.4.3	Aspettative	38
4.2.4.4	Ruoli del progetto	39
4.2.4.5	Gestione dei compiti	40
4.2.4.6	Gestione dei rischi	40
4.2.4.7	Metriche	41
4.2.5	Coordinamento	42
4.2.5.1	Scopo dell'attività	42
4.2.5.2	Descrizione della sezione	42
4.2.5.3	Aspettative	42
4.2.5.4	Comunicazioni	42
4.2.5.5	Riunioni	42
4.2.5.6	Norme generali	43
4.3	Gestione dell'infrastruttura	43
4.3.1	Scopo del processo	43
4.3.2	Descrizione della sezione	43
4.3.3	Aspettative	43
4.3.4	Strumenti	43
4.4	Miglioramento del processo	43
4.4.1	Scopo del processo	43
4.4.2	Descrizione della sezione	44
4.4.3	Aspettative	44
4.4.4	Norme generali	44
4.4.5	Metriche	44
4.5	Formazione del personale	44
4.5.1	Scopo del processo	44
4.5.2	Descrizione della sezione	44
4.5.3	Aspettative	44
4.5.4	Decisioni sulla formazione	44
A	Miglioramento di processo	45
A.1	SPICE - ISO/IEC 15504	45
A.2	Ciclo di Deming - PDCA	46
B	Miglioramento del software	47
B.1	Standard ISO/IEC 9126	47
B.2	Elenco delle metriche	47
B.2.1	Metriche per la qualità del processo	47
B.2.2	Metriche per la qualità di prodotto	48

1 Introduzione

1.1 Scopo del documento

Questo documento contiene le linee guida che tutti i componenti del gruppo sono tenuti a rispettare durante lo svolgimento del progetto, comprende quindi norme, tecnologie e strumenti che il gruppo *N.O.S* intende utilizzare per la realizzazione del prodotto finale. Le norme presenti potranno subire dei cambiamenti quali aggiunte, rimozioni e modifiche che saranno comunicate dal *Responsabile di progetto* a ciascun componente del gruppo.

1.2 Scopo generale del prodotto

Il progetto *Emporiolambda* ha come scopo quello di rendere disponibile un servizio e-commerce^G sfruttando però tutti i vantaggi di un'architettura Serverless^G:

- Gli sviluppatori potranno concentrare la propria attenzione sullo sviluppo del prodotto finale invece di focalizzarsi sulla gestione e sul funzionamento di server e di runtime, che siano nel cloud^G o in locale;
- Comodità nel costruire un insieme di chiamate asincrone^G che rispondono a diversi clienti contemporaneamente;
- Minori costi di sviluppo e di produzione;
- Semplicità nel suddividere il progetto in un insieme di microservizi^G.

1.3 Glossario

Al fine di rendere il documento più chiaro e leggibile si fornisce un *Glossario*. I termini che possono assumere un significato ambiguo sono indicati da una 'G' ad apice e fanno riferimento al documento *Glossario 3.0*.

1.4 Riferimenti

1.4.1 Normativi

- **Capitolato d'Appalto C2:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C2.pdf>

1.4.2 Informativi

- **Processi SW - Materiale del corso di Ingegneria del Software:**
 - <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L03.pdf>
 - https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- **Gestione di progetto - Materiale del corso di Ingegneria del Software:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L06.pdf>
- **Analisi dei Requisiti - Materiale del corso di Ingegneria del Software:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L07.pdf>
- **Progettazione - Materiale del corso di Ingegneria del Software:**
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L09.pdf>
- **Verifica - Materiale del corso di Ingegneria del Software:**
 - <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L15.pdf>
 - <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L16.pdf>
- **Standard ISO/IEC 12207:1995:**
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- **Diagrammi UML - Materiale del corso di Ingegneria del Software:**

- https://www.math.unipd.it/%7Eercardin/swea/2021/Diagrammi%20delle%20Classi_4x4.pdf
- https://www.math.unipd.it/%7Eercardin/swea/2021/Diagrammi%20dei%20Package_4x4.pdf
- https://www.math.unipd.it/%7Eercardin/swea/2021/Diagrammi%20di%20Attività_4x4.pdf
- https://www.math.unipd.it/%7Eercardin/swea/2021/Diagrammi%20di%20Sequenza_4x4.pdf
- https://www.math.unipd.it/%7Eercardin/swea/2021/Diagrammi%20Use%20Case_4x4.pdf
- **Standard ISO/IEC 15504:**
 - <http://museo.tecnoteca.it/sezioni/qualita/norme/iso15504>
 - <http://www.spice121.com/cms/en/about-spice-1-2-1.html>
- **Standard ISO/IEC 9126:**
http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf
- **PDCA - Ciclo di Deming:**
<https://www.ionos.it/startupguide/produttivita/pdca/>

2 Processi primari

2.1 Descrizione

I processi^G primari comprendono tutti i processi necessari per la gestione delle funzioni principali di un progetto durante il suo ciclo di vita, in particolare permettono di individuare e di organizzare le responsabilità del fornitore verso clienti, sviluppatori, operatori, altri fornitori e manutentori del prodotto richiesto.

Un progetto è in vita finché esiste un processo primario in esecuzione. In questa sezione vengono analizzati il **processo di fornitura** e il **processo di sviluppo**.

2.2 Processo di fornitura

2.2.1 Scopo del processo

Come indicato dallo standard^G ISO/IEC 12207:1995 il processo di fornitura è costituito dalle attività^G svolte dal fornitore al fine di comprendere e soddisfare le richieste del proponente.

Il processo di fornitura è costituito dalle seguenti attività:

- **Avvio:** il gruppo analizza i capitolati^G proposti e redige lo *Studio di fattibilità*;
- **Preparazione della risposta:** il gruppo si candida con una *lettera di presentazione* come fornitore per il capitolato^G scelto;
- **Pianificazione:** il gruppo pianifica il lavoro da svolgere e redige i documenti previsti dal progetto;
- **Esecuzione e controllo:** il gruppo segue quanto pianificato e individuato dai documenti;
- **Revisione e valutazione:** il gruppo esegue la verifica e la validazione del prodotto;
- **Consegna e completamento:** il gruppo è pronto al collaudo e alla consegna del prodotto realizzato.

2.2.2 Descrizione della sezione

La sezione comprende le norme principali per gestire correttamente i rapporti con il proponente *RedBabel* e i committenti *Prof. Vardanega Tullio* e *Prof. Cardin Riccardo*, oltre ad analizzare il documento che ha permesso al gruppo di scegliere quale progetto realizzare, ovvero lo *Studio di fattibilità*.

Le norme presenti riguardano quindi:

- Rapporti da mantenere con l'azienda proponente;
- Il materiale che il gruppo si impegna a presentare;
- Norme per la redazione efficace^G dello *Studio di fattibilità*.

2.2.3 Aspettative

Il gruppo *N.O.S* si impegna a mantenere un rapporto costante con il proponente *RedBabel* per poter procedere nella realizzazione del progetto coerentemente a quanto richiesto e stabilito con l'azienda.

2.2.4 Rapporti con l'azienda proponente RedBabel

Il gruppo utilizza esclusivamente i canali di comunicazione stabiliti con l'azienda proponente per le comunicazioni durante tutto il ciclo di vita del prodotto, come da accordi vengono utilizzati esclusivamente:

- Slack^G;
- Google Meet^G.

Il dialogo continuo con l'azienda permette in particolare di:

- Definire chiaramente i requisiti del proponente;
- Valutare l'andamento del lavoro;
- Proporre all'azienda eventuali alternative rispetto ad un determinato argomento;
- Chiarire eventuali dubbi emersi.

2.2.5 Materiale fornito

I documenti forniti all'azienda proponente e ai committenti *Prof. Vardanega Tullio* e *Prof. Cardin Riccardo* durante la realizzazione del progetto sono:

- **Analisi dei requisiti:** contiene la stesura di una dettagliata specifica dei requisiti che descrive le funzionalità del prodotto richiesto comprendendo i casi d'uso individuati durante lo studio del progetto;
- **Piano di progetto:** contiene la pianificazione preventiva dei tempi e delle attività, l'analisi dei rischi e il consuntivo di periodo, oltre alla data e ai costi previsti per la realizzazione del prodotto finale;
- **Piano di qualifica:** contiene gli obiettivi quantitativi di qualità^G fissati, l'analisi degli scostamenti e le misure correttive adottate;
- **Glossario:** contiene termini utilizzati dal gruppo nella documentazione che possono creare ambiguità;
- **Manuale utente e manutentore:** hanno lo scopo rispettivamente di guidare l'utente finale nell'utilizzo della piattaforma e permettere a sviluppatori esterni di accedere ad una descrizione approfondita dell'architettura del progetto;
- **Allegato tecnico:** documento di supporto alla PB^G che descrive approfonditamente l'architettura del sistema.

Alla documentazione è allegata la *Lettera di presentazione* con cui i membri del gruppo *N.O.S* formalizzano il loro impegno nel portare a termine il capitolato prescelto rispettando i requisiti minimi richiesti e la data di consegna. Alla documentazione si affianca:

- **Proof of Concept:** dimostratore funzionante che evidenzia come le tecnologie adottate dal gruppo possano essere integrate tra loro e utilizzate per lo sviluppo del prodotto richiesto.

Il prodotto software finale idoneo ad accettazione viene consegnato su un supporto CD-ROM/DVD.

2.2.6 Studio di fattibilità

Dopo la presentazione dei capitolati d'appalto avvenuta in data 2020_11_05, il gruppo si riunisce per dare una prima valutazione sui vari progetti proposti e, dopo i vari seminari tecnologici, per approfondire gli aspetti positivi e negativi di ciascuno. Individuato il capitolato di interesse gli *Analisti* svolgono un'ulteriore attività di analisi più approfondita e concludono la redazione dello *Studio di fattibilità*.

Il documento contiene le motivazioni che hanno spinto il gruppo a proporsi come fornitore del prodotto indicato e per ciascun capitolato riporta:

- Informazioni sul capitolato;
- Descrizione del capitolato;
- Scopo del capitolato;
- Tecnologie coinvolte;
- Vincoli;
- Aspetti positivi;
- Aspetti critici;
- Conclusioni.

2.2.7 Strumenti

Di seguito vengono riportati gli strumenti utilizzati durante il processo di fornitura.

- **GanttProject:** utilizzato per la realizzazione dei diagrammi di Gantt^G all'interno del documento *Piano di progetto 3.0*;
- **Microsoft Excel:** utilizzato per la realizzazione dei diagrammi all'interno del documento *Piano di progetto 3.0*.

2.3 Processo di sviluppo

2.3.1 Scopo del processo

Lo scopo del processo di sviluppo, come stabilito dallo standard ISO/IEC 12207:1995, è descrivere i compiti e le attività da svolgere per realizzare il prodotto software richiesto.

In questa sezione vengono analizzate le attività per:

- L'analisi dei requisiti;
- La progettazione del prodotto;
- La codifica del prodotto.

2.3.2 Descrizione della sezione

Nella sezione sono trattate le attività principali del processo di sviluppo con le relative norme stabilite dal gruppo N.O.S per la loro esecuzione.

Vengono analizzate le seguenti attività:

- **Analisi dei requisiti;**
- **Progettazione architetturale;**
- **Codifica del software.**

2.3.3 Aspettative

Le aspettative del gruppo riguardanti tale processo sono:

- Chiara individuazione dei requisiti del progetto;
- Chiara individuazione degli obiettivi del progetto;
- Chiara individuazione dei rischi del progetto;
- Chiara individuazione dei vincoli di design;
- Codifica del software coerente con i requisiti del proponente che supera i test di qualità previsti.

2.3.4 Analisi dei requisiti

2.3.4.1 Scopo dell'attività

L'attività di analisi dei requisiti viene svolta dagli *Analisti* incaricati per individuare i requisiti che il prodotto deve soddisfare. I requisiti possono essere esplicitamente richiesti dal proponente o individuati implicitamente tramite l'attività di analisi, possono essere individuati direttamente o indirettamente perché dipendono da altri requisiti.

L'individuazione dei requisiti consente di:

- Realizzare un prodotto software che soddisfi i bisogni del proponente;
- Facilitare le revisioni del codice;
- Fornire delle linee guida per le attività di test.

2.3.4.2 Descrizione della sezione

La sezione contiene le norme relative all'attività di analisi dei requisiti e del relativo documento *Analisi dei requisiti* che i componenti del gruppo, in particolare gli *Analisti*, devono seguire.

2.3.4.3 Aspettative

L'attività di analisi dei requisiti ha come obiettivo la redazione del documento di *Analisi dei requisiti* che comprende in modo formale tutti i requisiti che il prodotto software deve o è desiderabile soddisfi. Per assicurare il raggiungimento degli obiettivi di qualità previsti dal *Piano di qualifica* il gruppo intende:

- Coinvolgere frequentemente proponente e committente;
- Studiare prodotti simili a quello richiesto;
- Valutare i possibili scenari immedesimandosi nell'utente del prodotto;
- Individuare requisiti che possono migliorare la qualità del prodotto anche se non obbligatori;
- Assicurare che l'implementazione del prodotto soddisfi i requisiti individuati in modo coerente rispetto al loro significato.

2.3.4.4 Struttura del documento

La struttura del documento comprenderà:

- **Descrizione:** definisce le caratteristiche, i vincoli e gli obiettivi del prodotto;
- **Casi d'uso:** contiene tutte le informazioni utili per descrivere gli scenari che l'utente del prodotto può incontrare, comprendendo anche la loro rappresentazione attraverso i diagrammi UML;
- **Requisiti:** viene inserita una tabella contenente:
 - Codice identificativo del requisito;
 - Descrizione del requisito;
 - Fonte dalla quale è stato individuato il requisito.

2.3.4.5 Classificazione dei requisiti

Tutti i requisiti sono individuati da un codice identificativo univoco rappresentato nel seguente modo:

R[Tipologia][Rilevanza][Numero][_Codice]

- **Tipologia:** rappresenta il tipo di requisito, può assumere i seguenti valori:
 - **V:** requisito di **vincolo**;
 - **F:** requisito **funzionale**;
 - **P:** requisito **prestazionale**;
 - **Q:** requisito di **qualità**.
- **Rilevanza:** rappresenta l'utilità del requisito che va negoziata e concordata con il committente, può assumere i seguenti valori:
 - **O:** requisito **obbligatorio**, deve essere soddisfatto perché irrinunciabile per qualcuno degli stakeholder^G;
 - **D:** requisito **desiderabile**, non strettamente necessario ma aumenta la completezza del prodotto. Questi requisiti vengono negoziati con l'azienda *RedBabel*;
 - **Z:** requisito **opzionale**, relativamente utile anch'esso ed è contrattabile con l'azienda *RedBabel*.
- **Numero:** numero progressivo che parte da 1.
- **Codice:** opzionale, identifica il caso d'uso generico e gli eventuali sotto-casi ad esso associati, è rappresentato da:

[NumeroCasoBase](.NumeroSottoCaso)*

dove NumeroCasoBase e NumeroSottoCaso sono rappresentati da numeri progressivi.

I sotto-casi possono ramificarsi ulteriormente portandoli ad avere a loro volta altri sotto-casi.

Codice Requisito	Descrizione	Fonte
RFO1_1.1	Capitolato

Tabella 1: Esempio di classificazione requisito

2.3.4.6 Requisiti di processo

Dopo un confronto con il *Prof. Vardanega Tullio* si è ritenuto opportuno riportare i requisiti presenti nell'*Analisi dei requisiti* ritenuti da noi requisiti di processo all'interno di questo documento.

Descrizione	Fonte
La piattaforma deve essere sviluppata production-ready ^G .	Capitolato
Il codice sorgente deve essere realizzato con un sistema di versionamento ^G e caricato in una repository ^G su GitHub ^G .	Capitolato
Il codice sorgente deve essere sottoposto ad analisi statica attraverso lo strumento Typescript-eslint ^G .	Capitolato
Devono essere realizzati test di integrazione per verificare l'integrazione con i sottosistemi.	Capitolato
Devono essere realizzati test d'unità per verificare le singole componenti del prodotto.	Interna

Tabella 2: Requisiti di processo

2.3.4.7 Classificazione dei casi d'uso

Tutti i casi d'uso analizzati sono individuati da un codice identificativo univoco rappresentato nel seguente modo:

UC[NumeroCasoBase](.NumeroSottoCaso)*

dove:

- **NumeroCasoBase:** è costituito da un numero progressivo che indica il caso d'uso generico;
- **NumeroSottoCaso** è costituito da un numero progressivo opzionale che indica il sotto-caso d'uso del caso d'uso generico.

I sotto-casi possono avere a loro volta altri sotto-casi.

I casi d'uso analizzati comprenderanno:

- **Codice identificativo:** assegnato secondo quanto stabilito precedentemente;
- **Nome:** titolo assegnato al caso d'uso e indicato dopo il codice identificativo;
- **Rappresentazione grafica:** descrizione grafica del caso d'uso attraverso lo standard UML^G;
- **Descrizione:** breve descrizione testuale del caso d'uso;
- **Attori:** rappresentano gli utenti che hanno un'interazione con il sistema, si dividono in:
 - **Attori primari:** svolgono attivamente il caso d'uso;
 - **Attori secondari:** sono entità estranee al sistema che supportano gli attori primari nelle loro attività.
- **Precondizioni:**^G condizioni del sistema prima degli eventi che determinano il caso d'uso;
- **Postcondizioni:**^G condizioni del sistema dopo il verificarsi degli eventi che hanno determinato il caso d'uso;
- **Trigger:** opzionale, evento scatenante del caso d'uso;

- **Scenario principale:** elenco del flusso di eventi del caso d'uso;
- **Scenario alternativo:** opzionale, elenco delle azioni che si verificano allo scatenarsi di un evento non previsto rispetto allo scenario principale del caso d'uso;
- **Inclusioni:** opzionali, si ha inclusione quando un caso d'uso è incondizionatamente incluso nell'esecuzione del caso d'uso in esame;
- **Estensioni:** opzionali, si ha estensione quando l'esecuzione di un caso d'uso interrompe l'esecuzione del caso d'uso in esame;
- **Generalizzazioni:** opzionali, si ha generalizzazione quando si intende aggiungere o modificare caratteristiche base di un caso d'uso a un altro caso d'uso.

2.3.4.8 Qualità dell'analisi dei requisiti

Come indicato dallo standard IEEE 830-1998, la specifica deve essere:

- **Priva di ambiguità:** i requisiti devono rispettare chiaramente i bisogni dell'utente finale del prodotto;
- **Corretta:** i requisiti devono essere coerenti rispetto alle richieste degli utenti finali;
- **Completa:** i requisiti presenti permettono la completa comprensione del dominio del problema;
- **Verificabile:** deve essere possibile verificare il soddisfacimento dei requisiti da parte del prodotto;
- **Consistente:** i requisiti non devono essere contraddittori tra loro;
- **Modificabile:** i requisiti devono poter essere modificati senza perdita di consistenza e completezza;
- **Tracciabile:** l'origine dei requisiti è chiara e facilmente rintracciabile;
- **Ordinata per rilevanza:** i requisiti devono essere classificati secondo la loro rilevanza rispetto a quanto contrattato con il proponente.

2.3.4.9 Metriche

Di seguito vengono presentate le metriche utilizzate per garantire il controllo sulla qualità. Per una descrizione sullo standard di riferimento e un elenco completo di tutte le metriche applicate si rimanda a §B.1 dell'appendice.

- **MPR01: Soddisfacimento requisiti obbligatori**
Indica la percentuale di requisiti obbligatori soddisfatti nel momento in cui viene calcolato.
Viene calcolato nel seguente modo:

$$RO = \frac{ROC}{RO} * 100$$

dove:

- RO sta per **requisiti obbligatori**;
- ROC sta per **requisiti obbligatori coperti dall'implementazione**.

2.3.5 Progettazione

2.3.5.1 Scopo dell'attività

La progettazione ha lo scopo di definire quale sarà l'architettura logica del prodotto, viene svolta dai *Progettisti* prima dell'attività di codifica. Una buona architettura consente di:

- Ridurre la complessità del prodotto al fine di facilitare l'attività di codifica;
- Organizzare e ripartire le responsabilità di realizzazione;
- Realizzare un prodotto garantendo qualità e utilizzando il minor numero di risorse.

2.3.5.2 Descrizione della sezione

La sezione contiene le norme relative all'attività di progettazione che i componenti del gruppo, in particolare i *Progettisti*, devono seguire.

2.3.5.3 Aspettative

Il gruppo per questa attività si aspetta di definire l'architettura logica del prodotto rispettandone le caratteristiche di qualità. Per assicurare il raggiungimento degli obiettivi di qualità previsti dal *Piano di qualifica* il gruppo intende:

- Progettare perseguendo la correttezza per costruzione;
- Organizzare e ripartire le responsabilità di realizzazione considerando la natura a microservizi^G del prodotto da realizzare;
- Realizzare un prototipo del prodotto, il **Proof of Concept**^G, che comprenda le funzionalità principali per l'utente della piattaforma;
- Documentare quanto progettato attraverso i diagrammi UML;
- Progettare il prodotto sfruttando le qualità delle tecnologie da utilizzare;
- Progettare il prodotto considerando la facilità di utilizzo dello stesso per l'utente.

2.3.5.4 Qualità dell'attività di progettazione

Un'architettura di buona qualità ha come caratteristiche misurabili e osservabili oggettivamente:

- **Sufficienza:** capace di soddisfare tutti i requisiti indicati nell'*Analisi dei requisiti*;
- **Comprensibilità:** capace di essere capita da tutti gli stakeholders;
- **Modularità:** suddivisa in parti chiare e ben distinte;
- **Robustezza:** capace di gestire eventi non previsti causati dall'utente e dall'ambiente;
- **Flessibilità:** permette modifiche a costo contenuto al variare dei requisiti;
- **Riusabilità:** le sue parti possono essere utilizzate in altre applicazioni;
- **Affidabilità:** svolge quanto previsto;
- **Sicura:** capace di resistere a malfunzionamenti ed evitare possibili intrusioni.

Le sue componenti devono essere semplici, coese nel raggiungimento degli obiettivi, incapsulate e con un basso livello di accoppiamento. I *Progettisti* devono assicurare che l'architettura e le sue componenti presentino le caratteristiche descritte e superino dei test di qualità da loro definiti.

2.3.5.5 Periodi della progettazione

L'attività di progettazione si articola in due parti:

Progettazione architetturale Vengono definite le specifiche dell'architettura e delle componenti del prodotto, delle loro interazioni con le restanti parti del sistema e dei test di integrazione. Al termine di questo periodo di progettazione si ottiene la **Technology Baseline**^G del progetto che conterrà:

- **Proof of Concept:** primo eseguibile del sistema in grado di dimostrare che la tecnologia selezionata serve in modo efficace allo sviluppo del prodotto atteso;
- **Tecnologie utilizzate:** descrizione dettagliata delle tecnologie impiegate nello sviluppo del progetto, con particolare enfasi sui pregi e i difetti riscontrati;
- **Test di integrazione:** definizione dei test eseguiti per verificare che le varie componenti del sistema, una volta integrate insieme, interagiscano in modo corretto e in conformità con quanto richiesto dai requisiti;
- **Tracciamento delle componenti:** associazione tra requisiti e componenti che li soddisfano.

Progettazione di dettaglio Vengono definite le specifiche di dettaglio dell'architettura suddividendo il sistema fino ad arrivare a singole unità ben definite e realizzabili da un singolo programmatore. Al termine di questo periodo di progettazione si ottiene la **Product Baseline**^G del progetto che conterrà:

- **Design Pattern:** descrizione dei design pattern^G utilizzati nella definizione dell'architettura, per la soluzione progettuale a problemi ricorrenti riscontrati; ogni design pattern deve essere opportunamente descritto, con una spiegazione del suo significato, ed accompagnato da un diagramma che ne mostri la struttura;
- **Diagrammi UML:** diagrammi realizzati in linguaggio UML versione 2.0, utilizzati per rendere più chiare le soluzioni progettuali adottate;
- **Test di unità:** definizione dei test eseguiti per verificare che il funzionamento delle varie classi e metodi che implementano il sistema software sia corretto e conforme ai requisiti;
- **Tracciamento delle classi:** associazione tra requisiti e classi che li soddisfano.

2.3.5.6 Diagrammi UML 2.0

I diagrammi UML sono diagrammi realizzati in linguaggio UML, utilizzati per chiarire le scelte progettuali adottate e ridurre eventuali ambiguità. Di seguito vengono analizzati i diagrammi utilizzati.

Diagrammi delle classi Questi diagrammi descrivono il tipo degli oggetti che fanno parte di un sistema. Gli elementi di un diagramma di classe sono:

- **Nome:** nome della classe. Nel caso di un'interfaccia deve essere preceduto dalla direttiva **<<interface>>**;
- **Attributi:** la definizione di un attributo segue il formato:

Visibilità nome: tipo [molteplicità] = default [proprietà aggiuntive]

- **Visibilità:** visibilità dell'attributo rispetto ad altre classi, può essere:
 - * -: visibilità privata;
 - * +: visibilità pubblica;
 - * #: visibilità protetta;
 - * ~: visibilità di package.
- **Nome:** nome dell'attributo, se l'attributo è costante il nome deve essere scritto in maiuscolo;
- **Tipo:** tipo di dato dell'attributo;
- **Molteplicità:** occorrenze dell'attributo nella classe, il campo è opzionale;
- **Valore di default:** valore predefinito dell'attributo;

- **Proprietà aggiuntive:** eventuali proprietà ritenute importanti, il campo è opzionale.
- **Operazioni:** rappresentano le azioni eseguibili dalla classe, la definizione di una operazione segue il formato:

Visibilità nome [lista-parametri] : tipo-ritorno [proprietà aggiuntive]

- **Visibilità:** visibilità dell'operazione rispetto ad altre classi indicata come per gli attributi;
- **Nome:** nome dell'operazione;
- **Lista-parametri:** lista dei parametri dell'operazione;
- **Return-type:** tipo di ritorno dell'operazione;
- **Proprietà aggiuntive:** eventuali proprietà ritenute importanti, il campo è opzionale.

I diagrammi delle classi sono collegati tra loro tramite **relazioni di dipendenza**, la modifica di un elemento di un diagramma può cambiarne la definizione di un altro. Le relazioni di dipendenza previste sono:

- **Dipendenza:** gli oggetti di una classe A utilizzano brevemente gli oggetti di un'altra classe B, l'associazione esiste per un tempo limitato;



Figura 1: Dipendenza fra classi

- **Associazione:** la classe A contiene dei campi dati o delle istanze di un'altra classe B. Le molteplicità possibili sono:
 - **1:** A possiede un'istanza di B;
 - **0..1:** A possiede 0 o 1 istanze di B;
 - **0..*:** A possiede 0 o più istanze di B;
 - ***:** A possiede più istanze di B.



Figura 2: Associazione fra classi

- **Aggregazione:** la classe A possiede un riferimento ad un oggetto di un'altra classe B che può essere condiviso, l'aggregato B non ha senso di esistere senza l'aggregante A;

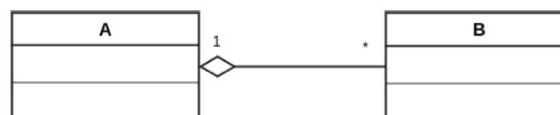


Figura 3: Aggregazione fra classi

- **Composizione:** la classe A possiede un oggetto di un'altra classe B, solo l'oggetto intero può creare e distruggere le sue parti;



Figura 4: Composizione fra classi

- **Generalizzazione:** la classe A generalizza un'altra classe B se ogni oggetto di B è anche un oggetto di A;
- **Subtyping:** la classe A implementa l'interfaccia B.

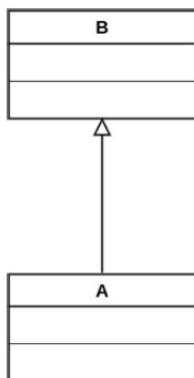


Figura 5: Generalizzazione fra classi

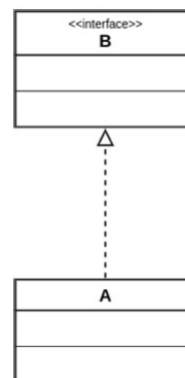


Figura 6: Subtyping fra classi

Diagrammi dei package Un package rappresenta un raggruppamento di un numero arbitrario di elementi UML in una unità di livello più alto, può quindi contenere classi (anche astratte), un altro package e interfacce. Ogni elemento di un package, identificativo di un namespace^G, può avere visibilità pubblica(+) o privata(-) e deve avere un nome completamente qualificato secondo lo schema:

package::package::.....classe

I diagrammi dei package documentano le dipendenze tra le classi che dovrebbero seguire tutte la stessa direzione evitando le dipendenze circolari.

Diagrammi delle attività Questi diagrammi descrivono la logica procedurale e i processi di business, aiutando a descrivere gli aspetti dinamici dei casi d'uso.

Gli elementi di un diagramma di attività sono:

- **Token:** vengono prodotti e consumati durante l'esecuzione;
- **Nodo iniziale:** rappresenta il punto d'inizio dell'esecuzione dell'attività, genera token;
- **Nodo di fine flusso:** rappresenta un punto di terminazione di un percorso di esecuzione, l'attività può continuare su altri percorsi;
- **Nodo finale:** rappresenta il punto di terminazione dell'esecuzione dell'attività, consuma token;



Figura 7: Nodo iniziale



Figura 8: Nodo fine flusso



Figura 9: Nodo finale

- **Activity:** rappresenta un'azione all'interno dell'attività, identificata da una breve descrizione;
- **Subactivity:** rappresenta una sotto-attività utilizzata per descrivere un'azione che ne comprende altre al suo interno, il suo diagramma viene fornito separatamente. Ogni sotto-attività è composta dall'input, dall'output e dalle azioni contenute.

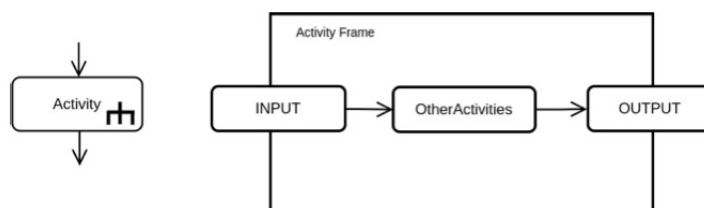


Figura 10: Attività che presenta una sotto-attività

- **Pin:** rappresenta un parametro prodotto o consumato da un'azione, va indicato il formato del parametro;

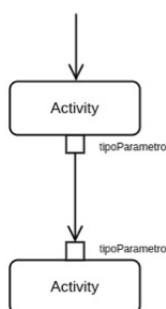


Figura 11: Attività che presenta un pin

- **Fork:** rappresenta un punto d'inizio di un'elaborazione parallela all'interno dell'attività, produce un token per ogni processo;
- **Join:** rappresenta un punto di sincronizzazione tra i processi paralleli, consuma i token in ingresso e ne genera solo uno;
- **Branch:** rappresenta un punto di decisione tra i possibili percorsi di esecuzione in base alla guardia associata;
- **Merge:** rappresenta un punto di unione dei diversi percorsi di esecuzione (non paralleli) generati in seguito ad un branch, il token viene solo instradato;

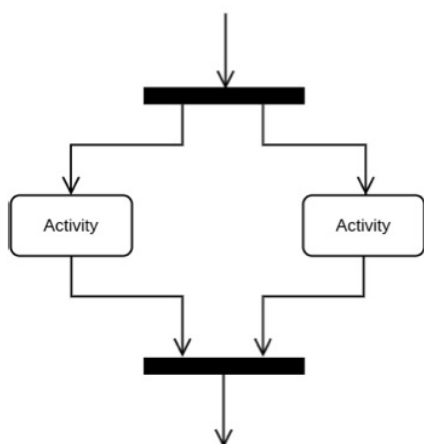


Figura 12: Fork e Join

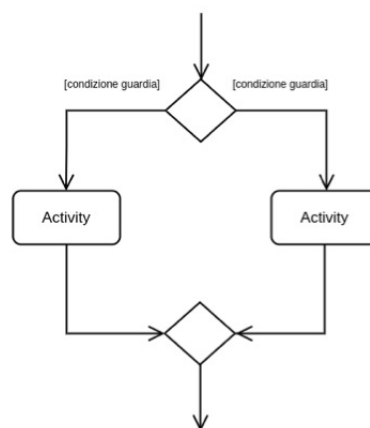


Figura 13: Branch e Merge

- **Segnale:** rappresenta un evento esterno, generato in modo non bloccante e catturato in modo bloccante, all'interno dell'attività;

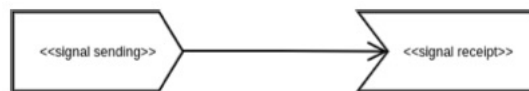


Figura 14: Segnali in un diagramma attività

- **Timeout:** rappresenta un'attesa bloccante all'interno dell'attività, di cui deve essere specificata la durata e l'unità di misura, o un evento ripetuto nel tempo;



Figura 15: Timeout in un diagramma attività

- **Swimlane:** fornisce una responsabilità all'esecuzione delle azioni all'interno di un'attività.

Diagrammi di sequenza I diagrammi di sequenza descrivono la collaborazione di un gruppo di oggetti che devono implementare collettivamente un comportamento. Gli elementi utilizzati in questi diagrammi sono i seguenti:

- **Partecipante:** entità che detiene il flusso di esecuzione del caso d'uso, è composto di due parti:
 - **Nome:** nome dell'entità partecipante;
 - **Barra di attivazione:** indica il periodo di tempo durante il quale il partecipante è attivo;
- **Messaggio:** rappresenta dati e operazioni scambiati tra partecipanti, può essere di una delle seguenti tipologie:
 - **Sincrono:** il chiamante rimane in attesa della risposta;
 - **Asincrono:**^G il chiamante non attende la risposta;
 - **Ritorno:** messaggio di ritorno riferito ad un precedente messaggio di chiamata;
 - **Creazione:** messaggio di creazione di un nuovo partecipante da parte del partecipante chiamante;
 - **Distruzione:** messaggio di distruzione di un partecipante da parte del partecipante chiamante.

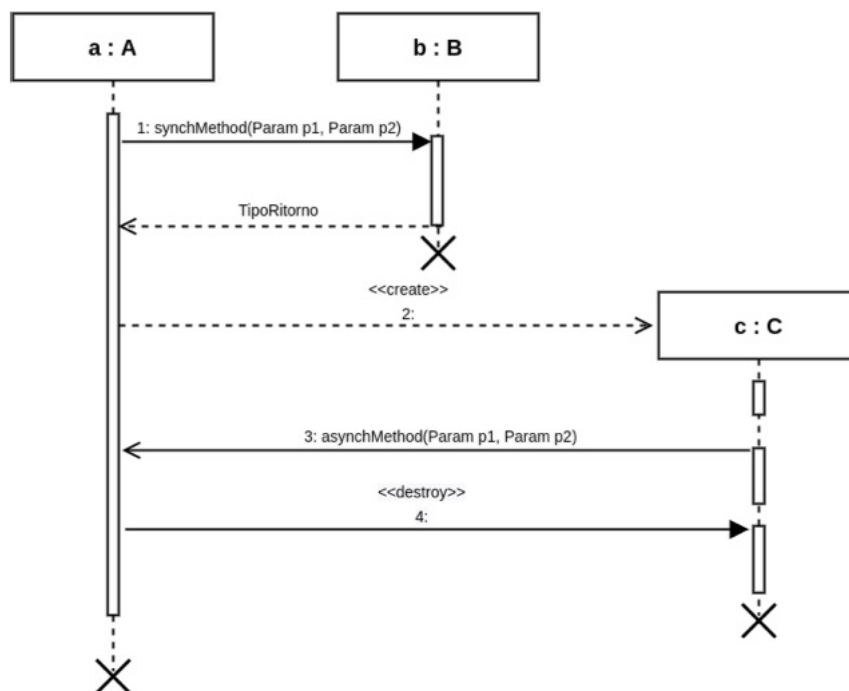


Figura 16: Esempio di diagramma di sequenza

Diagrammi dei casi d'uso Un caso d'uso è un'insieme di scenari (sequenze di azioni) che hanno in comune uno scopo finale per un utente. I diagrammi dei casi d'uso sono una rappresentazione grafica dei casi d'uso analizzati nel documento di *Analisi dei requisiti*. Vengono messi in evidenza:

- **Attori:** rappresentano tutto ciò che è esterno al sistema e interagisce con esso;
- **Use Case:** rappresentano le attività associate al sistema.

Le eventuali relazioni tra gli use case del sistema rappresentate in questi diagrammi sono:

- **Inclusione:** si ha quando vi sono funzionalità comuni fra più casi d'uso, lo use case B è incondizionatamente incluso nell'esecuzione dello use case A;

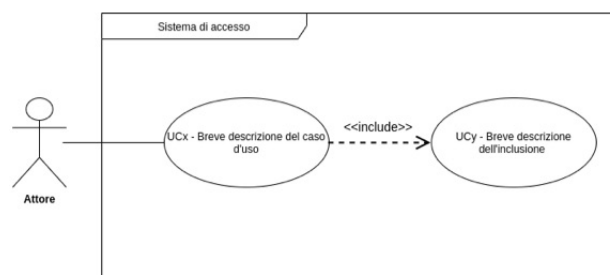


Figura 17: Inclusione casi d'uso

- **Estensione:** si ha se ogni istanza dello use case A esegue lo use case B in modo condizionato, l'esecuzione di B interrompe A;

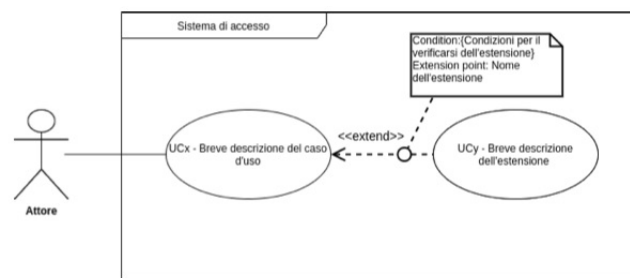


Figura 18: Estensione casi d'uso

- **Generalizzazione:** si ha quando si vogliono aggiungere o modificare caratteristiche base, in particolare si ha che:
 - L'attore A è generalizzazione dell'attore B se B condivide almeno le funzionalità di A;
 - I casi d'uso figli aggiungono funzionalità rispetto ai padri o ne modificano il comportamento.

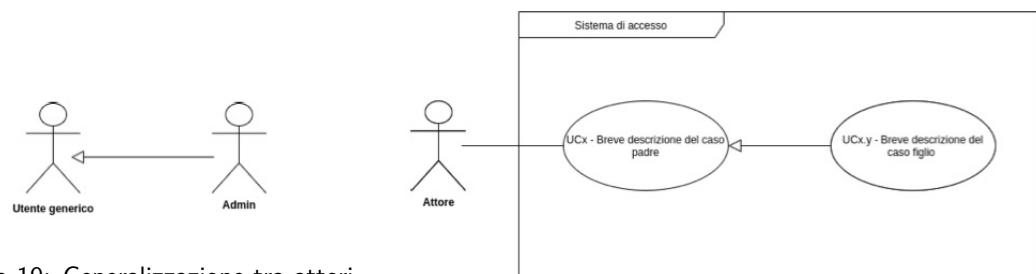


Figura 19: Generalizzazione tra attori

Figura 20: Generalizzazione Casi d'uso

2.3.5.7 Metriche

Di seguito vengono presentate le metriche utilizzate per garantire il controllo sulla qualità. Per una descrizione sullo standard di riferimento e un elenco completo di tutte le metriche applicate si rimanda a §B.1 dell'appendice.

- **MPDS03: Tempo di caricamento delle schermate;**
Il valore stabilisce la velocità con cui le varie schermate del prodotto software vengono caricate e mostrate all'utente, viene calcolato attraverso uno strumento automatico;
- **MPDS04: Facilità di utilizzo;**
Il valore stabilisce la semplicità per un utente di raggiungere quanto cercato valutando il numero di click medio per raggiungere il contenuto desiderato;
- **MPDS05: Facilità di apprendimento**
Il valore stabilisce la semplicità per un utente di raggiungere quanto cercato all'interno del sito calcolando la media dei minuti necessari per raggiungere le pagine ricercate;
- **MPDS06: Profondità della gerarchia**
Il valore risultante viene calcolato individuando la profondità della gerarchia delle schermate rispetto alla principale.

2.3.6 Codifica software

2.3.6.1 Scopo dell'attività

L'attività di codifica svolta dai *Programmatore* ha lo scopo di trasformare in codice quanto previsto dall'attività di progettazione svolta dai *Progettisti*, procedendo all'effettiva realizzazione del prodotto software richiesto.

2.3.6.2 Descrizione della sezione

La sezione tratta le norme da seguire per ogni linguaggio di programmazione utilizzato durante lo svolgimento del progetto.

2.3.6.3 Aspettative

Con le norme individuate il gruppo si aspetta la realizzazione di codice uniforme. Per assicurare il raggiungimento degli obiettivi di qualità previsti dal *Piano di qualifica* il gruppo intende:

- Utilizzare i linguaggi sfruttando adeguatamente le loro capacità;
- Ridurre il numero di righe di codice che possono causare errori;
- Redarre il codice affinché risulti essere chiaro nel tempo e facilmente comprensibile anche da altri programmatori.

2.3.6.4 Qualità della codifica

La codifica è di qualità se:

- Il codice è facilmente leggibile;
- I costrutti del linguaggio sono utilizzati in modo chiaro e coerente;
- La compilazione non presenta errori fatali o potenziali.

Queste caratteristiche sono in grado di agevolare manutenzione, verifica e validazione e di conseguenza migliorare la qualità di prodotto.

2.3.6.5 Convenzioni generiche

Vengono di seguito riportate le norme generiche stabilite per la codifica.

Intestazione I file sorgenti consegnati presentano la seguente intestazione inserita in un blocco di commento.

Nomenclatura Tutte le classi, i metodi, le variabili devono avere un nome univoco, esplicativo e scritto in lingua inglese, in particolare:

- Per i nomi di cartelle, file e classi viene seguita la convenzione **CamelCase**;
- I nomi di variabili e metodi hanno iniziale minuscola, se composti da più parole la prima lettera delle parole successive alla prima è maiuscola;
- Le costanti vengono scritte in maiuscolo.

Parentesi I blocchi di codice vanno inseriti tra parentesi graffe, anche se il blocco è vuoto o costituito da una sola riga di codice.

Verbosità Una riga di codice deve essere lunga al massimo 140 caratteri. Se possibile è desiderabile definire metodi brevi evitando la ricorsione.

2.3.6.6 Metriche

Di seguito vengono presentate le metriche utilizzate per garantire il controllo sulla qualità. Per una descrizione sullo standard di riferimento e un elenco completo di tutte le metriche applicate si rimanda a §B.1 dell'appendice.

- **MPDS01: Completezza dell'implementazione**

L'indice stabilisce la completezza del prodotto software realizzato nel momento in cui viene calcolato rispetto a tutti i requisiti previsti dall'*Analisi dei requisiti 2.0*.

Viene calcolato nel seguente modo:

$$CI = 1 - \frac{FNI}{FI} * 100$$

dove:

- **FNI** indica il numero di funzionalità non implementate;
- **FI** indica il numero di funzionalità implementate.

▪ **MPDS02: Densità errori**

Rappresenta la percentuale di righe di codice che possono causare errori.
Viene calcolato nel seguente modo:

$$DE = \frac{RB}{RTOT} * 100$$

dove:

- **RB** indica il numero di linee di codice che possono causare imprevisti o comportamenti diversi da quelli desiderati;
- **RTOT** indica il numero di linee di codice totali.

▪ **MPDS07: Facilità di comprensione**

Il valore indica la facilità di comprensione del codice in base al numero di righe di commento presenti.
Viene calcolato nel seguente modo:

$$FC = \frac{LCOD}{LCOM} * 100$$

dove

- **LCOM** indica il numero di linee di commento;
- **LCOD** indica il numero di linee di codice.

▪ **MPDS08: Semplicità delle funzioni**

Il valore stabilisce la semplicità di un metodo basandosi sul numero di parametri a lui necessari.

▪ **MPDS09: Semplicità delle classi**

Il valore stabilisce la semplicità di una classe basandosi sul numero di metodi al suo interno.

2.3.7 Strumenti

Di seguito sono riportati gli strumenti utilizzati durante il processo di sviluppo.

- **Diagrams.net:** utilizzato per la realizzazione dei diagrammi UML^G all'interno del documento *Analisi dei requisiti 2.0*.

3 Processi di supporto

3.1 Descrizione

I processi^G di supporto comprendono tutti i processi che nell'eseguire una specifica funzione forniscono supporto ad un altro processo che li esegue, contribuendo al successo e alla qualità^G di un progetto.

I processi di supporto analizzati in questa sezione sono:

- Documentazione;
- Gestione della configurazione;
- Gestione della qualità;
- Verifica;
- Validazione;
- Gestione dei cambiamenti.

3.2 Documentazione

3.2.1 Scopo dell'attività

Quando si realizza un progetto è di fondamentale importanza documentare processi e attività^G significative per lo sviluppo dello stesso.

La documentazione permette di:

- Disciplinare ciò che verrà fatto durante il ciclo di vita del software;
- Garantire che tutte le informazioni d'interesse siano facilmente accessibili agli stakeholders^G;
- Assicurare che quanto previsto dalla documentazione sia rispettato per garantire la qualità attesa.

3.2.2 Descrizione della sezione

La sezione comprende le norme relative alla documentazione associata al progetto che tutti i componenti devono seguire nella redazione dei documenti di loro competenza.

3.2.3 Aspettative

L'obiettivo del gruppo è di ottenere dei documenti coerenti e validi dal punto di vista tipografico e formale. I documenti redatti dal gruppo devono:

- Essere facilmente aggiornabili e modificabili;
- Contenere solo ciò che è essenziale;
- Essere facili da leggere.

Per assicurare il raggiungimento degli obiettivi di qualità previsti dal *Piano di qualifica* il gruppo intende:

- Utilizzare un metodo di versionamento^G per il tracciamento delle versioni;
- Rendere disponibile solo versioni di documenti verificati e quindi consistenti;
- Individuare regole di impaginazione comune;
- Creazione di un glossario per chiarire il significato di termini che possono risultare ambigui;
- Assicurare che tutto ciò che sia di interesse degli stakeholders^G sia presente;
- Controllare attentamente l'ortografia e il livello di comprensione di quanto scritto.

3.2.4 Ciclo di vita di un documento

Tutti i documenti redatti sono sottoposti a queste attività:

- **Creazione:** il documento viene creato secondo il template presente nella repository^G;
- **Stesura:** il documento viene redatto dai componenti incaricati, tutti i documenti devono contenere:
 - Registro delle modifiche;
 - Indice dei contenuti.
- **Revisione:** le sezioni del documento vengono regolarmente revisionate da uno o più componenti del gruppo diversi dai redattori delle stesse;
- **Approvazione:** il *Responsabile di progetto* stabilisce la validità del documento che può in seguito essere rilasciato.

3.2.5 Classificazione dei documenti

I documenti vengo suddivisi in:

- **Documenti Pubblici:** comprendono i documenti di interesse per il gruppo, il proponente e il committente. Sono documenti pubblici:
 - **Glossario:** elenco ordinato di termini utilizzati dal gruppo nei loro documenti e che possono presentare ambiguità nel loro significato;
 - **Studio di fattibilità:** analisi dei capitolati^G proposti, con le valutazioni da parte dei membri del gruppo;
 - **Piano di progetto:** espone la pianificazione delle attività di progetto e relative previsioni rispetto a impiego orario dei membri, preventivo e consuntivi di periodo;
 - **Piano di qualifica:** presenta i criteri di valutazione della qualità utilizzati dal gruppo;
 - **Analisi dei requisiti:** contiene tutti i requisiti e le caratteristiche del prodotto finale individuati dal gruppo.
- **Documenti Interni:** comprendono i documenti di interesse principalmente per i componenti del gruppo. Sono documenti interni:
 - **Norme di progetto:** raccolta delle norme stabilite dal gruppo che tutti i componenti devono seguire durante la realizzazione del progetto;
 - **Verbali:** si suddividono a loro volta in:
 - * **Interni:** contengono un resoconto sintetico relativo alle riunioni svolte dal gruppo;
 - * **Esterni:** contengono un resoconto relativo alle riunioni del gruppo con il proponente e/o i committenti.

3.2.6 Nomenclatura dei documenti

Tutti i documenti ad eccezione dei verbali seguiranno il seguente schema per la nomenclatura:

[NomeDocumento]_v[X].[Y]

dove:

- **[NomeDocumento]** corrisponde al nome ufficiale del documento, non deve presentare spazi e viene utilizzata la notazione **CamelCase**.
- **v[X].[Y]** dove v sta per "versione" e X, Y rispettano quanto descritto in §3.3.6

Per quanto riguarda la nomenclatura dei verbali si segue il seguente schema:

V[I/E]_[YYYY]_[MM]_[DD]

dove:

- **V**: indica che il documento è un verbale;
- **[I/E]**: indica se il verbale è interno o esterno;
- **[YYYY]_[MM]_[DD]**: indica la data in cui si è svolto l'incontro indicata come anno_mese_giorno.

3.2.7 Directory dei documenti

Ogni documento viene inserito in una directory il cui nome coincide con il NomeDocumento dello stesso, posizionata a sua volta secondo la tipologia del documento contenuto nella directory **DocumentiInterni** o **DocumentiPubblici**.

3.2.8 Struttura dei documenti

Tutti i documenti contengono prima delle pagine con il contenuto previsto:

- **Frontespizio**: la prima pagina di ogni documento comprende in ordine i seguenti elementi:
 - **Logo del gruppo**;
 - **Nome del documento**;
 - **Nome del gruppo e del progetto**;
 - **Email del gruppo**;
 - **Informazioni sul documento**, che includono:
 - * **Versione**;
 - * **Approvatore**;
 - * **Redattori**;
 - * **Verificatori**;
 - * **Uso** se interno o pubblico;
 - * **Distribuzione**, cioè i destinatari del documento.
 - **Descrizione** breve del contenuto del documento.
- **Registro delle modifiche**: sotto forma di tabella viene presentato un diario delle modifiche contenente i cambiamenti effettuati al documento durante il suo ciclo di vita e le varie verifiche effettuate. Ciascuna voce della tabella riporta:
 - **Versione** del documento dopo la modifica/verifica;
 - **Data** della modifica/verifica;
 - **Nominativo** del componente che ha effettuato la modifica/verifica;
 - **Ruolo** del componente che ha effettuato la modifica/verifica;
 - **Descrizione** della modifica/verifica effettuata.

Per garantire la consistenza del registro le sezioni sono etichettate con il comando \LaTeX **label**, le sezioni che vengono eliminate sono indicate con il loro titolo.

- **Indice**: dopo il registro delle modifiche viene inserito un indice dei contenuti utile a chi utilizza il documento per orientarsi velocemente all'interno dello stesso.
- **Corpo dei documenti**: dopo l'indice dei contenuti è presente il corpo del documento. Gli argomenti sono divisi in sezioni e sottosezioni raggruppati in modo coerente e coeso. Il corpo dei verbali deve contenere le seguenti sezioni:
 - **Informazioni generali**, che includono:
 - * **Luogo** in cui si è svolta la riunione;
 - * **Data** in cui si è svolta la riunione;

- * **Ora**, comprendente orario di inizio e di fine;
- * **Partecipanti**, membri del gruppo ed eventualmente committenti e/o proponente;
- * **Segretario**, componente del gruppo che ha redatto il verbale.
- **Ordine del giorno**: argomenti principali di cui si è trattato;
- **Resoconto**: per ogni argomento trattato è presente una breve descrizione della discussione avvenuta;
- **Registro delle decisioni**: vengono presentate sotto forma di elenco le decisioni prese dal gruppo.

3.2.9 Formattazione e norme tipografiche

3.2.9.1 Template

Il gruppo utilizza per la stesura dei documenti il linguaggio \LaTeX . Al fine di mantenere una coerenza nella formattazione dei documenti vengono utilizzati due template permettendo così l'automatizzazione delle scelte tipografiche.

I template utilizzati sono:

- **Stilelettera**: template esclusivamente per le *lettere di presentazione*;
- **Stiletemplate**: template per tutti i restanti documenti.

3.2.9.2 Termini di glossario

I termini il cui significato potrebbe essere ambiguo sono individuati con un apice ^G alla fine della parola, salvo che il suo significato non venga subito spiegato. Tutti i termini da glossario sono stati inseriti nel documento *Glossario*. Di ogni termine solamente la prima occorrenza per ogni sezione va evidenziata con l'apice ^G.

3.2.9.3 Stile del testo

Gli stili di testo adottati nei documenti sono:

- **grassetto**: per titoli, sottotitoli, nome degli oggetti negli elenchi puntati, termini importanti secondo il *Redattore*;
- **corsivo**: per i nomi propri di persona, nomi dei ruoli, nome del progetto, nome dei documenti;
- **maiuscolo**: per i nomi dei file, delle directory e dei processi (convenzione **CamelCase**).

Ogni volta che si utilizza il nome di un documento come titolo o in una voce di elenco seguita da una descrizione, tale nome deve seguire solo la prima convenzione senza utilizzare il corsivo.

3.2.9.4 Formattazione di date e orari

Per l'indicazione delle date e degli orari si segue quanto sancito dallo standard^G ISO 8601.

Le date vengono indicate secondo il seguente formato:

[YYYY]_[MM]_[DD]

dove:

- **[YYYY]** corrisponde all'anno;
- **[MM]** corrisponde al mese;
- **[DD]** corrisponde al giorno.

Gli orari vengono indicati secondo il seguente formato:

[HH]:[MM]

dove:

- **[HH]** rappresenta l'ora;
- **[MM]** rappresenta i minuti.

3.2.9.5 Formattazione elementi grafici

La sezione definisce le norme per gli elementi grafici all'interno dei documenti.

- **Immagini:** le figure presenti nei documenti vanno centrate rispetto al testo e accompagnate da una didascalia coerente;
- **Grafici UML:** i grafici in linguaggio UML^G sono inseriti come immagini;
- **Tabelle:** le tabelle vanno accompagnate da una didascalia opportuna. Nella didascalia viene indicato l'identificativo

Tabella [X]

dove [X] indica il numero assoluto della tabella all'interno del documento, segue poi il testo della didascalia. A questa prassi fanno eccezione le tabelle del registro delle modifiche.

3.2.10 Metriche

Di seguito vengono presentate le metriche utilizzate per garantire il controllo sulla qualità. Per una descrizione sullo standard di riferimento e un elenco completo di tutte le metriche applicate si rimanda a §B.1 dell'appendice.

- **MPR02: Indice di Gulpease**

L'indice permette di stabilire la difficoltà di lettura di un testo. Viene calcolato nel seguente modo:

$$GULP = 89 + \frac{300 \times \#frasi - 10 \times \#lettere}{\#parole}$$

L'indice può assumere i seguenti valori:

- **GULP < 80:** indica una leggibilità difficile per un utente con licenza elementare;
 - **GULP < 60:** indica una leggibilità difficile per un utente con licenza media;
 - **GULP < 40:** indica una leggibilità difficile per un utente con licenza superiore.
- **MPR03: Indice di correttezza ortografica**

Indica la correttezza ortografica in un documento. Viene calcolato nel seguente modo:

$$CORT = \# \text{ numero di errori ortografici}$$

Un risultato maggiore di zero indica la presenza di errori nel documento.

3.2.11 Strumenti

Di seguito sono riportati gli strumenti utilizzati durante la redazione della documentazione.

- **LaTeX;**
- **GanttProject:** utilizzato per la realizzazione dei diagrammi di Gantt^G all'interno del documento *Piano di progetto 3.0*;
- **Microsoft Excel:** utilizzato per realizzare i diagrammi all'interno dei documenti;
- **Diagrams.net:** utilizzato per la realizzazione dei diagrammi UML^G all'interno del documento *Analisi dei requisiti 2.0*.

3.3 Gestione della configurazione

3.3.1 Scopo dell'attività

La gestione della configurazione ha lo scopo di gestire ordinatamente e sistematicamente la produzione dei documenti e del codice.

3.3.2 Descrizione della sezione

La sezione presenta le norme che il gruppo intende seguire per il controllo della documentazione e del codice.

3.3.3 Aspettative

Con queste norme il gruppo intende assicurare una corretta gestione dei documenti e del codice rendendo sistematica la loro produzione e garantendo la chiara individuazione delle versioni prodotte.

3.3.4 Identificazione di configurazione

Ogni configuration item^G ha un'identità unica che ne consente la chiara individuazione. Per una trattazione precisa si rimanda a §3.2.6.

3.3.5 Controllo di baseline

Per garantire riproducibilità e tracciabilità viene definita una successione di baseline^G associate a una serie di milestone^G specifiche per obiettivi di avanzamento e coerenti con la strategia di progetto. Data l'importanza di controllare lo svolgimento del progetto il numero delle milestone sarà superiore al numero delle revisioni.

3.3.6 Controllo di versione

3.3.6.1 Codice di versione per documenti e software

Il codice di una versione è formato:

[X].[Y]

dove:

- **[X]** corrisponde ad una versione del documento approvata dal *Responsabile di progetto* e pronta al rilascio, la numerazione parte da 0;
- **[Y]** corrisponde ad una versione del documento verificata in tutte le sue parti, inizia da 0 e riparte da questo valore ad ogni incremento di X;

Il controllo di versione si appoggia su un repository creato in Github^G la cui struttura viene spiegata nella sezione seguente.

3.3.7 Struttura della repository

La repository del gruppo *N.O.S* è accessibile all'indirizzo

<https://github.com/NotOnlyStudents>

L'utilizzo di questo strumento permette a ciascun componente di lavorare sui configuration items^G senza rischio di sovrascritture consentendo allo stesso tempo di controllare l'andamento del lavoro. All'interno del repository viene divisa la documentazione dal codice software.

3.3.7.1 Repository per la documentazione

Il repository per la documentazione è composto da diversi **branch**:

- **master** su cui vengono riportati i documenti approvati dal *Responsabile di progetto* e pronti nella loro versione definitiva ad essere presentati;
- **develop** ramo in cui vengono inserite la versioni dei documenti verificati ma non ancora approvati;
- rami dedicati ai singoli documenti in sviluppo non verificati.

All'interno del repository troviamo le cartelle così organizzate:

- **Immagini**: contiene le immagini utilizzate all'interno dei documenti;
- **Utilita**: contiene file di utilità per la produzione di tutti i documenti;
- **DocumentiInterni**: contiene le cartelle dei documenti di interesse per il gruppo e marginale per proponente e committenti;
- **DocumentiPubblici**: contiene le cartelle dei documenti di interesse per il gruppo, per il proponente e per i committenti.

3.3.7.2 Repository per il Poc

I file con il codice relativi al back end^G e front end^G del PoC^G da realizzare sono stati inseriti rispettivamente in:

<https://github.com/NotOnlyStudents/poc-be>

<https://github.com/NotOnlyStudents/poc-fe>

3.3.7.3 Modifiche al repository

Nel caso fosse necessario modificare file all'interno del ramo master è necessaria una **pull request** con conseguente approvazione da almeno un altro elemento del gruppo.

3.4 Gestione della qualità

3.4.1 Scopo dell'attività

L'attività ha lo scopo di assicurare al proponente che il prodotto realizzato soddisfi i requisiti e la qualità previsti e stabiliti con il proponente.

3.4.2 Descrizione della sezione

La sezione presenta il processo utilizzato dal gruppo per gestire il sistema qualità^G, indicando procedure e strumenti utili che i componenti devono seguire.

3.4.3 Aspettative

Il gruppo si aspetta da questo processo di conseguire la qualità del prodotto prevista secondo le aspettative del proponente attraverso un'azione preventiva. Il documento *Piano di qualifica 3.0* individua quanto stabilito indicando gli obiettivi di qualità da raggiungere e le metriche^G da utilizzare per la valutazione della qualità.

3.4.4 Istanziamento di un processo

Perseguendo la qualità attraverso un'azione preventiva, il gruppo deve compiere le proprie scelte considerando le caratteristiche di qualità ricercate. Le seguenti norme devono essere considerate con l'istanziamento di ogni processo:

- Ogni processo deve puntare a perseguire un unico obiettivo;
- Ad ogni istanziazione di un processo devono essere analizzati i rischi connessi e prevista una pianificazione del lavoro per il loro contrasto;
- Ad ogni istanziazione di un processo le risorse vengono assegnate considerando gli altri processi già in esecuzione;
- Ogni processo deve avere una durata definita.

3.4.5 Svolgimento di un processo

Durante lo svolgimento di un processo, le attività vengono svolte perseguendo il raggiungimento degli obiettivi individuati all'interno del *Piano di qualifica 3.0* e le seguenti caratteristiche di qualità di prodotto:

- **Funzionalità;**
- **Efficienza;**
- **Usabilità;**
- **Affidabilità;**
- **Manutenibilità.**

Vengono utilizzati degli standard di riferimento descritti nelle sezioni §A.1, §A.2 e §B.1 dell'appendice, inoltre periodicamente vengono calcolate delle metriche^G utili per valutare l'andamento delle attività e adattare di conseguenza mirando al miglioramento continuo. All'interno del *Piano di qualifica* è possibile visualizzare i limiti di qualità che il gruppo intende raggiungere per assicurare la qualità prevista e l'andamento dei risultati ottenuti.

3.4.6 Classificazione degli obiettivi

Gli obiettivi di qualità presenti nel *Piano di qualifica 3.0* vengono classificati secondo il seguente formato:

O[Categoria][D/S][X]

dove:

- **O** indica che si tratta di un obiettivo di qualità;

- **Categoria** può assumere i seguenti valori:
 - **PR**: l'obiettivo riguarda un processo;
 - **PD**: l'obiettivo riguarda un prodotto.
- **D/S** indicano rispettivamente se l'obiettivo riguarda un documento o un prodotto software, è presente solo per gli obiettivi relativi ai prodotti;
- **X** è un numero progressivo che inizia da 1.

3.4.7 Classificazione delle metriche

Le metriche utilizzate per la valutazione della qualità presenti nel *Piano di qualifica 3.0* vengono classificate secondo il seguente formato:

M[Categoria][D/S][X]

dove:

- **M** indica che si tratta di una metrica di qualità;
- **Categoria** può assumere i seguenti valori:
 - **PR**: la metrica riguarda un processo;
 - **PD**: la metrica riguarda un prodotto.
- **D/S** indicano rispettivamente se l'obiettivo riguarda un documento o un prodotto software, è presente solo per gli obiettivi relativi ai prodotti;
- **X** è un numero progressivo che inizia da 1.

Per una descrizione approfondita delle metriche utilizzate si rimanda a §B.2 dell'appendice.

3.5 Verifica

3.5.1 Scopo dell'attività

L'attività di verifica ha lo scopo di accertare che quanto eseguito durante il periodo preso in considerazione non abbia introdotto errori e quindi che quanto realizzato sia corretto e completo.

3.5.2 Descrizione della sezione

La sezione descrive come avviene l'attività di verifica da parte del gruppo *N.O.S* per la documentazione e il codice.

3.5.3 Aspettative

Utilizzando le norme presenti il gruppo si aspetta di ottimizzare e uniformare l'attività di verifica. L'attività di verifica viene attuata attraverso l'**analisi statica** e l'**analisi dinamica**.

3.5.4 Analisi statica

L'analisi statica studia la documentazione e il codice accertando che siano conformi alle regole stabilite, non vi siano difetti e presentino le proprietà desiderate. Il *Verificatore* seguirà i seguenti metodi:

- **Walkthrough:**^G procede ad una lettura critica del prodotto in esame alla ricerca di difetti;
- **Inspection:**^G procede ad una lettura critica del prodotto in esame, focalizzando la ricerca su problemi che si presentano più frequentemente indicati nella lista di controllo all'interno del *Piano di qualifica*.

3.5.4.1 Verifica della documentazione

Ogniquale volta si redige una sezione o si apporta una modifica al documento, viene effettuata la verifica. Il *Verificatore* incaricato procede nel seguente modo:

- Rilegge la sezione, nel caso emergano sostanziali errori si valuta l'aggiornamento della lista di controllo;
- Successivamente viene applicato il metodo di Inspection concentrandosi sulle criticità presenti nella lista di controllo;
- Se riscontra problemi logici, si accorda con il redattore del documento per la correzione;
- Chiude la pull request collegata e automaticamente l'issue associata passa in "VERIFIED".

3.5.4.2 Verifica del codice

Il *Verificatore* incaricato procede nel seguente modo:

- Rilegge interamente il codice scritto al fine di:
 - Accertare che il codice esegua nella sequenza specificata e sia ben strutturato;
 - Localizzare codice non raggiungibile;
 - Identificare parti di codice che possano non terminare;
 - Accertare che nessun cammino acceda a variabili prive di valore;
 - Rilevare anomalie;
 - Provare la correttezza del codice sorgente rispetto ai requisiti;
 - Assicurare che sia stato seguito quanto scritto nelle *Norme di progetto* in §2.3.6.5.

Nel caso emergano sostanziali errori si valuta l'aggiornamento della lista di controllo;

- Successivamente ad ogni verifica viene applicato il metodo di Inspection concentrandosi sulle criticità presenti nella lista di controllo;
- Se riscontra problemi, si accorda con il redattore del documento per la correzione.
- Chiude la pull request collegata e automaticamente l'issue associata passa in "VERIFIED".

3.5.5 Analisi dinamica

L'analisi dinamica procede a verificare il programma durante la sua esecuzione. Per procedere a questa attività vengono utilizzati GitHub Actions e dei test ripetibili e automatizzati con specifici:

- **Ambiente di esecuzione;**
- **Input e Output;**
- **Procedure;**

3.5.5.1 Tipologia di test effettuati

I test permettono di valutare se il programma svolge i tasks per cui è stato sviluppato. Il *Piano di qualifica* specifica quali e quanti test effettuare.

Test d'unità I test d'unità^G consistono nell'isolare e testare la parte più piccola di software, chiamata appunto unità. Lo scopo è di valutare la logica interna del codice d'unità in modo da poterne stabilire il corretto funzionamento.

Test d'integrazione Applicati alle componenti della progettazione architettuale, hanno lo scopo di valutare che l'aggregazione delle parti del sistema non introduca errori nel software prodotto.

Test di regressione Questi test sono necessari a verificare che la modifica di una parte P di S non causi errori in P, in S o in una parte in relazione con essi. Le modifiche apportate non devono pregiudicare le funzionalità già verificate.

Test di sistema Vengono effettuati per valutare che il prodotto soddisfi tutti i requisiti software previsti.

3.5.5.2 Procedura di verifica

Il *Verificatore* incaricato procede nel seguente modo:

- Esegue tutti i test previsti;
- Se riscontra problemi, si accorda con il redattore del codice per la correzione.
- Chiude la pull request collegata e automaticamente l'issue associata passa in "VERIFIED".
- Avverte i redattori incaricati ad aggiornare il *Piano di qualifica* con i risultati ottenuti.

3.6 Validazione

3.6.1 Scopo dell'attività

La validazione stabilisce se il prodotto soddisfa i compiti per cui è stato creato, in particolare permette di garantire che quanto richiesto e stabilito da *RedBabel* sia soddisfatto.

3.6.2 Descrizione della sezione

La sezione descrive la procedura seguita per la validazione di quanto viene svolto durante il progetto.

3.6.3 Aspettative

Il gruppo si aspetta di procedere alla validazione dei documenti in modo chiaro.

3.6.4 Validazione dei documenti

Il *Responsabile di progetto* procede nel seguente modo:

- Se non ancora creata, crea l'issue in GitHub e la sposta in "IN PROGRESS";
- Se riscontra problemi, avverte il redattore del documento e il verificatore. L'issue torna in "TO DO" in attesa delle modifiche. Quando queste vengono apportate si torna al primo punto;
- Sposta l'issue in "VERIFIED" approvando il documento;
- Avverte i redattori incaricati di aggiornare il *Piano di qualifica* con i risultati ottenuti.

3.6.5 Validazione del codice

Il *Verificatore* incaricato procede nel seguente modo:

- Se non ancora creata, crea l'issue in GitHub;
- Sposta l'issue in "IN PROGRESS";
- Esegue i test descritti in §3.5.5.1 così da garantire un software pronto al rilascio;
- Sposta l'issue in "VERIFIED" approvando il documento.

3.6.6 Piano di qualifica e test

All'interno del *Piano di qualifica* sono inseriti tutti i test pianificati e implementati dai *Progettisti*, eseguiti durante il progetto.

Per ogni test devono essere specificati:

- Entità sottoposte a verifica;
- Task e obiettivi della validazione;
- Membri che hanno eseguito i test;
- Resoconto dei risultati ottenuti.

3.6.7 Nomenclatura dei test

Ogni test viene descritto con:

- Codice identificativo;
- Descrizione;

Il codice identificativo viene così assegnato:

T[Tipo][ID]

dove:

- **T** indica test;
- **Tipo**: assume i seguenti valori:
 - **I**: test d'integrazione;
 - **U**: test d'unità;
 - **S**: test di sistema;
 - **V**: test di validazione;
- **ID**: rappresenta un numero progressivo.

3.7 Gestione dei cambiamenti

3.7.1 Scopo dell'attività

L'attività ha lo scopo di garantire una corretta gestione dei problemi che si possono verificare durante il progetto.

3.7.2 Descrizione della sezione

La sezione descrive la prassi generale che il gruppo intende seguire nel caso di complicazioni o malfunzionamenti.

3.7.3 Aspettative

Il gruppo si aspetta di affrontare tempestivamente attraverso quanto previsto dall'attività, gli eventuali problemi che si possono presentare.

3.7.4 Norme generali

Come descritto in §4.2.4.5, ogni componente gestisce autonomamente i compiti che gli spettano. Periodicamente viene effettuata l'attività di verifica secondo quanto previsto da §3.5. È di fondamentale importanza tracciare tutte le osservazioni che ci vengono fatte notare da committenti e proponente per poter correggere tempestivamente eventuali problemi. Tutti i cambiamenti che verranno apportati ad attività o prodotti sono riportati nel *Piano di qualifica*.

3.7.5 Denominazione dei cambiamenti

Per facilitare il tracciamento dei cambiamenti, vengono classificati secondo il seguente schema:

C[P/A][Attività/Prodotto][ID]

dove:

- **C** indica cambiamento;
- **P/A** indicano rispettivamente:
 - **P** se riguarda il prodotto;
 - **A** se riguarda un'attività.
- **Attività/Prodotto** possono assumere i seguenti valori:
 - **AN** se riguarda l'attività di analisi;
 - **PG** se riguarda l'attività di progettazione;
 - **CD** se riguarda l'attività di codifica;
 - **DO** se riguarda l'attività di documentazione;
 - **CG** se riguarda l'attività di configurazione;
 - **QU** se riguarda l'attività di gestione della qualità;
 - **VE** se riguarda l'attività di verifica;
 - **PN** se riguarda l'attività di pianificazione.
 - **D** se riguarda un prodotto documentale;
 - **C** se riguarda un prodotto di codifica.
- **ID** è un numero incrementale che parte da 1.

4 Processi organizzativi

4.1 Descrizione

I processi^G organizzativi comprendono tutti i processi utilizzati per definire e gestire una struttura costituita dal personale e dai processi del ciclo di vita del prodotto, al fine di migliorarne continuamente la struttura e i processi stessi. In questa sezione vengono analizzati:

- **Gestione di processo;**
- **Gestione dell'infrastruttura;**
- **Miglioramento del processo;**
- **Formazione del personale.**

4.2 Gestione di processo

4.2.1 Scopo del processo

Come stabilito dallo standard^G ISO/IEC 12207:1995, comprende le attività^G e i compiti generici per la gestione dei rispettivi processi.

4.2.2 Descrizione della sezione

La sezione definisce le norme stabilite dal gruppo per la pianificazione e il coordinamento del lavoro dei componenti del team.

4.2.3 Aspettative

Le aspettative del gruppo per questo processo sono:

- Pianificare in modo ragionevole le attività da seguire;
- Gestire i componenti del gruppo assegnando ruoli e compiti;
- Monitorare l'andamento del progetto.

4.2.4 Pianificazione

4.2.4.1 Scopo dell'attività

Lo scopo dell'attività di pianificazione è di definire un metodo di lavoro stabilendone la fattibilità e assicurandone la sua esecuzione.

4.2.4.2 Descrizione della sezione

Nella sezione vengono stabiliti compiti e ruoli di progetto, gestione dei rischi.

4.2.4.3 Aspettative

Con l'attività di pianificazione il gruppo intende definire il metodo di lavoro che viene seguito durante la realizzazione del progetto. Per assicurare il raggiungimento degli obiettivi di qualità^G previsti dal *Piano di qualifica* il gruppo intende:

- Pianificare le attività da svolgere con precise scadenze;
- Introdurre più milestone^G nel tempo rispetto al numero di revisioni;
- Stabilire ed eventualmente riassegnare le risorse necessarie alla realizzazione delle attività considerando le risorse utilizzate in altre attività;
- Valutare periodicamente il progresso delle attività ed eventualmente modificare quanto pianificato.

4.2.4.4 Ruoli del progetto

Tutti i componenti del gruppo devono ricoprire tutti i ruoli almeno una volta durante lo svolgimento del progetto. I ruoli previsti dal progetto sono elencati di seguito.

Responsabile di progetto Coordina e organizza il lavoro e rappresenta il gruppo presso il proponente e il committente per tutta la durata del progetto.

I suoi compiti principali sono:

- Predisporre l'attività di pianificazione e assicurarne l'esecuzione;
- Approvare il rilascio di qualunque prodotto di progetto, sia software che documentale;
- Redigere **organigramma** e *Piano di progetto*;
- Coordinare le attività, le risorse e i componenti del gruppo.

Amministratore di progetto Gestisce, controlla e cura gli strumenti che il gruppo utilizza per il proprio lavoro.

I suoi compiti principali sono:

- Gestire i problemi legati alla gestione dei processi;
- Gestire la documentazione del progetto;
- Controlla versioni e configurazioni del prodotto;
- Redigere le *Norme di progetto*.

Analista Svolge tutte le attività necessarie all'analisi dei requisiti.

I suoi compiti principali sono:

- Studiare il dominio del problema;
- Definire tutti i requisiti del prodotto richiesto;
- Redigere *Studio di fattibilità* e *Analisi dei requisiti*.

Progettista È responsabile della progettazione del prodotto partendo da quanto svolto dall'*Analista*.

I suoi compiti principali sono:

- Seguire lo sviluppo del prodotto;
- Influire sulle scelte tecniche e tecnologiche;
- Redigere la documentazione dell'architettura del prodotto.

Programmatore È responsabile delle attività di codifica del prodotto e delle componenti necessarie per la sua verifica e validazione.

I suoi compiti principali sono:

- Codificare quanto previsto dal *Progettista*;
- Codificare i componenti ausiliari di supporto alla verifica e alla validazione;
- Redigere i manuali.

Verificatore È responsabile dell'attività di verifica.

I suoi compiti principali sono:

- Controllare che le attività di processo non abbiano causato errori;
- Redigere la parte retrospettiva del *Piano di qualifica* che descrive l'esito delle verifiche e delle prove effettuate.

4.2.4.5 Gestione dei compiti

Durante lo sviluppo del progetto spetta al singolo componente stabilire in piena autonomia quali sono i suoi compiti a seconda del proprio ruolo. Per permettere di gestire i compiti in modo agevole è stato scelto di utilizzare l'ITS^G fornito da GitHub^G: il repository^G usato dal gruppo per il versionamento^G del progetto. Il componente del gruppo in base al proprio ruolo crea delle issue^G che rappresentano dei singoli compiti, che assegna a se stesso o ad altri componenti in seguito a riunioni e secondo accordi, comprendenti le seguenti informazioni:

- **Titolo:** nome del compito da eseguire;
- **Descrizione:** descrizione dettagliata del compito da eseguire;
- **Assegnatari:** persone a cui compete lo svolgimento del compito;
- **Project:** cruscotto di progetto in cui il compito sarà monitorato;
- **Milestone:** data entro la quale lo svolgimento del compito deve essere completato.

Ogni issue attraversa degli stati che permettono di monitorare l'avanzamento nello svolgimento del compito che essa rappresenta, questi sono:

- **TO DO:** compito da svolgere;
- **IN PROGRESS:** compito in svolgimento;
- **VERIFIED:** compito svolto.

Il membro del gruppo dopo aver aperto l'issue relativa all'attività da svolgere, apre un branch di lavoro e sposta l'issue in "IN PROGRESS" quando inizia il compito. Quando ha concluso l'attività invia una pull request collegando la issue relativa, a questo punto è possibile procedere all'attività di verifica. Nel caso in cui l'operazione di verifica vada a buon fine, la pull request viene chiusa insieme alle relative criticità associate. In caso contrario, alla presenza di grosse mancanze all'interno del documento stesso, il *Verificatore* si accorda con il componente del gruppo che ha modificato il documento e lo corregge secondo le indicazioni ricevute e basandosi sulla lista di controllo.

4.2.4.6 Gestione dei rischi

Tutti i rischi che si possono presentare durante la realizzazione del progetto vengono inseriti all'interno del *Piano di progetto* redatto dal *Responsabile di progetto*. Viene seguita la seguente procedura:

- Individuazione di nuovi problemi e monitoraggio dei rischi già individuati;
- Aggiunta dei nuovi rischi nel *Piano di progetto*;
- Ridefinizione se necessario delle strategie di gestione.

Per consentire l'identificazione univoca ogni rischio viene individuato con un codice secondo il seguente schema:

R[Tipologia][Numero]

dove:

- **Tipologia:** identifica uno dei seguenti tipi di rischio:
 - **T:** rischio legato alla tecnologia;
 - **G:** rischio legato ai membri del gruppo;
 - **O:** rischio legato all'organizzazione;
 - **R:** rischio legato ai requisiti.
- **Numero:** incrementale a partire da 1, per ogni tipologia si riparte da 1.

Nel *Piano di progetto 3.0* per ogni rischio è presente una tabella indicante:

- **Codice identificativo e titolo:** secondo quanto indicato precedentemente e un breve titolo;

- **Descrizione;**
- **Conseguenze** causate dal verificarsi del rischio;
- **Probabilità di occorrenza;**
- **Pericolosità;**
- **Precauzioni;**
- **Piano di contingenza.**

4.2.4.7 Metriche

Di seguito vengono presentate le metriche utilizzate per garantire il controllo sulla qualità. Per una descrizione sullo standard di riferimento e un elenco completo di tutte le metriche applicate si rimanda a §B.1 dell'appendice.

- **MPR04: Budget at Completion**

Rappresenta il valore previsto per la realizzazione del progetto secondo quanto previsto nel *Piano di progetto 1.0.0*.

- **MPR05: Actual Cost of Work Performed**

Corrisponde alla somma di tutte le spese sostenute dal gruppo fino al momento in cui l'indice viene calcolato, il limite superiore dipende da quanto stabilito nel Preventivo^G presente nel *Piano di progetto 1.0.0*.

- **MPR06: Budget Cost of Work Performed**

Valore con virgola effettivo del prodotto ottenuto fino al momento in cui l'indice viene calcolato.

- **MPR07: Budget Cost of Work Scheduled**

Valore con virgola corrispondente alla spesa pianificata per l'attività di progetto alla data corrente, reperibile nel preventivo presente nel *Piano di progetto 1.0.0*.

- **MPR08: Schedule Variance**

Indica se si è in linea, in anticipo o in ritardo rispetto alla schedulazione delle attività di progetto pianificate nella baseline^G.

Viene calcolato nel seguente modo:

$$SV = BCWP - BCWS$$

dove:

- **BCWP** sta per **budgeted cost of work performed**;
- **BCWS** sta per **budgeted cost of work scheduled**.

In base al risultato ottenuto abbiamo:

- **SV>0**: il gruppo sta producendo con maggior velocità rispetto quanto pianificato;
- **SV=0**: il gruppo sta producendo secondo quanto pianificato;
- **SV<0**: il gruppo sta producendo con minor velocità rispetto quanto pianificato, è necessaria una revisione di quanto pianificato.

- **MPR09: Budget Variance**

Indica se il gruppo rientra nel budget di spesa previsto alla data corrente.

Viene calcolato nel seguente modo:

$$BV = BCWS - ACWP$$

dove:

- **BCWS** sta per **budgeted cost of work scheduled**;
- **ACWP** sta per **actual cost of work performed**.

Un valore positivo segnala che i costi sostenuti rientrano nel budget preventivato.

4.2.5 Coordinamento

4.2.5.1 Scopo dell'attività

L'attività di coordinamento ha lo scopo di definire come vengono gestite le comunicazioni all'interno del gruppo N.O.S e tra il gruppo e soggetti esterni.

4.2.5.2 Descrizione della sezione

La sezione presenta le norme che il gruppo segue per le comunicazioni e le riunioni.

4.2.5.3 Aspettative

L'aspettativa del gruppo per questa attività è di stabilire metodi chiari di comportamento da seguire durante lo svolgimento del progetto.

4.2.5.4 Comunicazioni

I membri del gruppo sono tenuti a seguire quanto indicato per tutta la durata del progetto.

Comunicazioni interne I membri utilizzano per le comunicazioni interne il gruppo Telegram^G appositamente creato. Per tutte le comunicazioni importanti viene preferito utilizzare Zoom effettuando quindi delle video-chiamate che permettono un feedback immediato.

Comunicazioni esterne I soggetti esterni con cui il gruppo può comunicare sono:

- **I committenti** *Prof. Vardanega Tullio e Prof. Cardin Riccardo*;
- **L'azienda RedBabel** rappresentata da *Milo Ertola e Alessandro Maccagnan*;
- **Gruppi concorrenti** che lavorano allo stesso progetto *Emporio λ mbda*;

Per le comunicazioni con il proponente si rimanda a §2.2.4.

Le comunicazioni con i committenti avvengono esclusivamente tramite l'email del gruppo.

`nos.unipd@gmail.com`

ed eventualmente tramite video-conferenze su Zoom.

Le comunicazioni con i gruppi concorrenti avvengono in modo informale tramite Telegram.

4.2.5.5 Riunioni

In caso di necessità, i membri possono proporre di organizzare una riunione per discutere di eventuali problemi o dubbi. Ogni incontro dovrà essere fissato in accordo con i partecipanti in base alle loro disponibilità, se un componente è impossibilitato a partecipare per un periodo eccessivamente esteso accetta quanto stabilito dagli altri componenti. Gli incontri possono avvenire tra i soli membri del gruppo o tra il gruppo e i committenti/proponenti. Tutto ciò che viene stabilito durante le riunioni viene inserito in un verbale da un segretario incaricato. Per la struttura dei verbali si rimanda a §3.2.8.

Incontri interni Agli incontri interni partecipano solamente i membri del gruppo in modalità virtuale tramite video-chiamata Zoom di gruppo. Se e quando la situazione attuale legata all'emergenza sanitaria migliorerà e sarà possibile accedere liberamente alle aule dell'Università di Padova, il gruppo può decidere se svolgere le riunioni in presenza presso le aule di Torre Archimede - Dipartimento di Matematica "Tullio Levi Civita" (Via Trieste 63, 35121 Padova(PD)). Tra un incontro e un altro viene creato un documento condiviso in Google Drive nel quale ogni componente può inserire gli argomenti che vuole trattare nel prossimo incontro, questi costituiranno l'ordine del giorno della riunione.

Incontri esterni Negli incontri esterni, assieme ai membri del gruppo, sono coinvolti anche uno o più rappresentanti dell'azienda proponente. Gli incontri esterni con l'azienda proponente avvengono secondo gli accordi (vedi §2.2.4). Se si necessita di discutere con i committenti il *Responsabile di progetto* ha il compito di contattarli e organizzare l'incontro secondo le disponibilità del *Prof. Vardanega Tullio* e del *Prof. Cardin Riccardo*.

4.2.5.6 Norme generali

I componenti del gruppo si impegnano a:

- Essere il più possibile reperibili;
- Informare tempestivamente il gruppo in caso di problemi di salute e/o famigliari che non consentono il corretto svolgimento dei propri compiti;
- Aggiornare il *Responsabile di progetto* a intervalli regolari rispetto il lavoro in svolgimento;
- Informare tempestivamente il gruppo se si hanno difficoltà nello svolgimento dei compiti di propria competenza.

4.3 Gestione dell'infrastruttura

4.3.1 Scopo del processo

Il processo di gestione dell'infrastruttura ha lo scopo di stabilire tutti gli strumenti utilizzati per gestire l'infrastruttura necessaria per tutti gli altri processi.

4.3.2 Descrizione della sezione

La sezione indica quali strumenti vengono utilizzati dal gruppo per il coordinamento e la pianificazione del lavoro durante la realizzazione del progetto.

4.3.3 Aspettative

I componenti del gruppo utilizzano quanto previsto di seguito per operare senza ambiguità e coerentemente con quanto stabilito.

4.3.4 Strumenti

Gli strumenti utilizzati dal gruppo sono:

- **Zoom:** strumento utilizzato per le video-conferenze tra i membri del gruppo;
- **Telegram:** strumento di messaggistica istantanea basato su cloud^G, utilizzato dal gruppo *N.O.S* per le comunicazioni tra i vari membri;
- **Gmail:** strumento open source^G di posta elettronica, utilizzato per gestire la casella e-mail del gruppo;
- **Google Drive:** servizio web basato su cloud per la memorizzazione e condivisione di file, utilizzato dal gruppo per condividere file utili ai propri membri;
- **Google Calendar:** servizio web basato su cloud utilizzato per avere accesso ad un calendario in cui segnare incontri e impegni personali;
- **Slack:**^G strumento collaborativo aziendale, utilizzato per inviare messaggi in modo istantaneo ai membri di un team di lavoro. Come indicato in §2.2.4 viene utilizzato per eventuali comunicazioni con il proponente *RedBabel*;
- **Google Meet:**^G strumento utilizzato per le video-conferenze tra i membri del gruppo e il proponente *RedBabel*;
- **GanttProject:** utilizzato per la pianificazione e la realizzazione dei diagrammi di Gantt^G.

4.4 Miglioramento del processo

4.4.1 Scopo del processo

Il processo di miglioramento ha lo scopo di stabilire, valutare, misurare, controllare e migliorare il ciclo di vita del software.

4.4.2 Descrizione della sezione

La sezione analizza le tre attività principali del processo di miglioramento, ovvero:

- **Istituzione del processo;**
- **Valutazione del processo;**
- **Miglioramento del processo.**

4.4.3 Aspettative

Il gruppo si pone come aspettativa di questo processo la pianificazione del lavoro nel rispetto del principio di miglioramento continuo. Per assicurare il raggiungimento degli obiettivi di qualità previsti dal *Piano di qualifica* il gruppo intende:

- Correggere tempestivamente quanto segnalatoci da committenti e proponente;
- Mantenere traccia dei cambiamenti effettuati come indicato in §3.7;
- Rendere stabili nel tempo i miglioramenti conseguiti;
- Ridurre al minimo necessario il tempo di gestione delle issue da svolgere.

4.4.4 Norme generali

All'interno del *Piano di qualifica* vengono inseriti gli obiettivi e le metriche di qualità per valutare l'andamento delle attività di ogni processo. Qualora si presentassero problemi, il membro del gruppo deve avvertire tempestivamente gli altri componenti del team e si procederà ad una eventuale modifica di quanto pianificato. Periodicamente vengono calcolate le metriche di qualità previste.

4.4.5 Metriche

Di seguito vengono presentate le metriche utilizzate per garantire il controllo sulla qualità. Per una descrizione sullo standard di riferimento e un elenco completo di tutte le metriche applicate si rimanda a §B.1 dell'appendice.

- **MPR10: Indice di risoluzione dei problemi**
L'indice calcola la velocità con cui le issue vengono risolte, viene calcolata la media dei numeri di giorni che intercorrono tra l'apertura e la chiusura per ogni issue.

4.5 Formazione del personale

4.5.1 Scopo del processo

Il processo di formazione consente di pianificare la preparazione dei componenti del gruppo rispetto a tecnologie e strumenti utilizzati durante la realizzazione del progetto.

4.5.2 Descrizione della sezione

La sezione comprende le norme stabilite dal gruppo per l'autonoma formazione dei componenti.

4.5.3 Aspettative

Attraverso questo processo il gruppo intende acquisire le competenze necessarie per poter realizzare il prodotto con la qualità stabilita in modo efficace^G ed efficiente^G.

4.5.4 Decisioni sulla formazione

I membri del gruppo provvedono ad approfondire le proprie conoscenze e a colmare le proprie lacune sulle tecnologie e sugli strumenti che vengono utilizzati durante tutti i processi. I membri più esperti hanno il compito di condividere le proprie conoscenze per velocizzare il processo di formazione secondo il principio di miglioramento continuo.

A Miglioramento di processo

A.1 SPICE - ISO/IEC 15504

L'ISO/IEC 15504, noto come progetto **SPICE** (Software Process Improvement and Capability Evaluation), ha come obiettivo quello di sviluppare uno standard^G che consenta valutazioni omogenee di organizzazioni software allo scopo di:

- Identificare i punti forti, i punti deboli e i rischi delle metodologie utilizzate;
- Individuare le situazioni nelle quali le metodologie utilizzate permettono di raggiungere in maniera efficace^G gli obiettivi prefissati.

Per effettuare la classificazione dei diversi processi, viene valutato e attribuito loro un livello di **Capability**, cioè la capacità di essere cognitivamente capace di raggiungere il suo scopo. Ogni livello comprende degli attributi sui quali effettuare la valutazione. I livelli considerati con i loro attributi sono i seguenti:

- **Livello 0 Incomplete:** il processo non è implementato e non è in grado di raggiungere gli obiettivi;
- **Livello 1 Performed:** il processo è implementato e raggiunge gli obiettivi ma non è sottoposto a controlli costanti ai fini di correzione e miglioramento. Attributi associati:
 - **Performance di processo (PP):** numero degli obiettivi raggiunti.
- **Livello 2 Managed:** il processo è gestito tramite pianificazione, controllo e correzione, l'output raggiunge gli obiettivi fissati. Attributi associati:
 - **Gestione delle performance (PM):** grado di organizzazione degli obiettivi fissati;
 - **Gestione del prodotto di lavoro (WPM):** grado di organizzazione dei prodotti rilasciati.
- **Livello 3 Established:** esiste un insieme di standard per il processo. Attributi associati:
 - **Definizione di processo (PDEF):** grado di adesione del processo agli standard;
 - **Rilascio di processo (PDEP):** grado di misura della garanzia di rilascio con reperibilità.
- **Livello 4 Predictable:** il processo è istanziato coerentemente rispetto ai limiti previsti, è quantitativamente misurato per assicurarne il mantenimento delle performance. Attributi associati:
 - **Misurazioni di processo (PME):** misura con quanta efficacia^G le metriche possono essere applicate al processo;
 - **Controllo di processo (PC):** grado di predicibilità dei risultati delle valutazioni.
- **Livello 5 Innovating:** il processo migliora attraverso feedback quantitativi e gli standard sono adattati di conseguenza. Attributi associati:
 - **Innovazione di processo (PI):** misura quanto i cambiamenti attuati nel processo siano innovativi e positivi;
 - **Ottimizzazione di processo (PO):** misura quanto la curva di miglioramento sia lineare.

Ogni attributo viene valutato e ad esso viene assegnato uno dei seguenti livelli, a seconda del grado di soddisfazione dello stesso:

- **N not achieved:** 0%-15%, il processo non ha implementato l'attributo o presenta gravi lacune;
- **P partially achieved:** 16%-50%, il processo ha implementato l'attributo in modo sistematico ma risulta migliorabile o poco predicibile;
- **L largely achieved:** 51%-85%, il processo ha ampiamente implementato l'attributo ma il suo valore è poco uniforme rispetto alle altre parti del processo;
- **F fully achieved:** 86%-100%, il processo ha implementato completamente l'attributo ed è uniforme in ogni sua parte.

A.2 Ciclo di Deming - PDCA

Il ciclo **PDCA** è un metodo iterativo con l'obiettivo di stabilire un modello per il miglioramento continuo dei processi ed essere garanzia di qualità^G efficiente^G e continuativa. A tal fine si individuano quattro passaggi:

- **Plan:** pianificazione, si esamina la situazione attuale al fine di determinare esattamente come raggiungere obiettivi specifici di miglioramento definendo attività, scadenze, responsabilità e risorse da impiegare;
- **Do:** implementazione, si procede a piccoli passi per realizzare quanto previsto durante la pianificazione;
- **Check:** verifica, vengono raccolti i risultati delle azioni di miglioramento ottenuti e valutati rispetto agli obiettivi prefissati;
- **Act:** azione, viene consolidato nel way of working quanto ha prodotto effettivi miglioramenti mentre vengono individuati dei modi per migliorare ciò che non ha apportato valore.

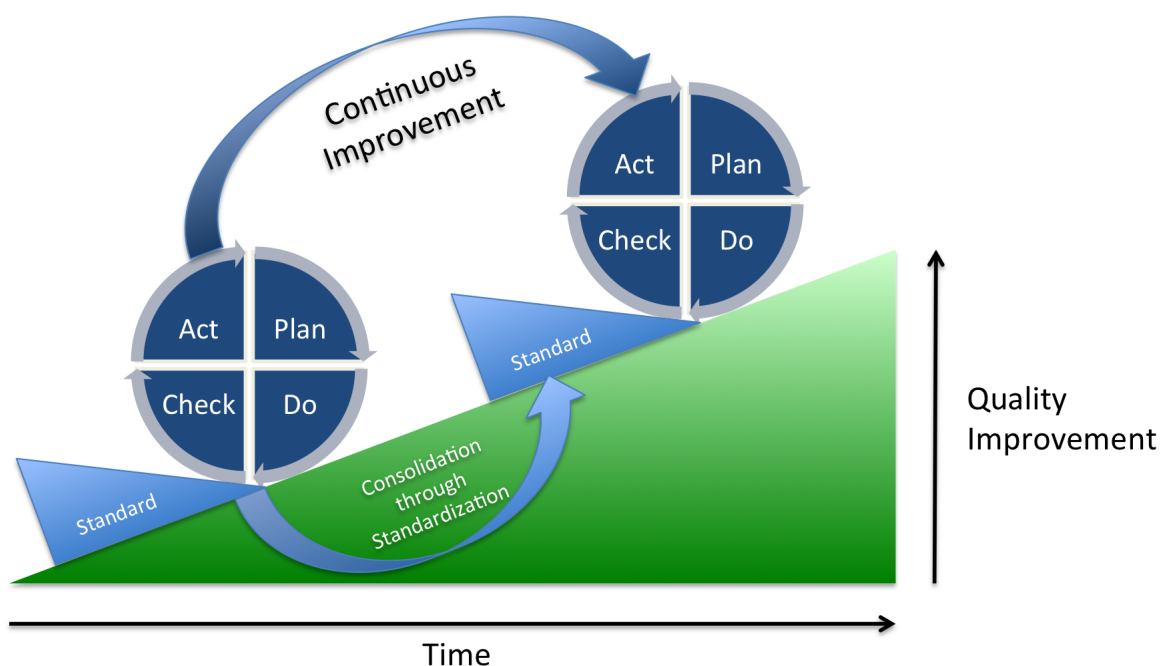


Figura 21: Ciclo di Deming

B Miglioramento del software

B.1 Standard ISO/IEC 9126

L'ISO/IEC 9126 è uno standard internazionale per la valutazione della qualità del software, le norme presenti definiscono le caratteristiche che la determinano e propongono metriche per la misurazione. Il modello di qualità software stabilito dallo standard considera sei caratteristiche principali da soddisfare, ovvero:

- **Funzionalità;**
- **Affidabilità;**
- **Efficienza;**
- **Usabilità;**
- **Manutenibilità;**
- **Portabilità.**

Le caratteristiche elencate possono essere misurate da **metriche esterne**, **metriche interne** e **metriche per la qualità d'uso**.

Metriche esterne

Misurano i comportamenti del prodotto software rilevabili dai test, dall'operatività e dall'osservazione durante la sua esecuzione, in funzione degli obiettivi stabiliti.

Metriche interne

Si applicano alla documentazione e al prodotto non eseguibile, ad esempio il codice sorgente, durante la progettazione e la codifica. Queste metriche misurano attributi interni del software e forniscono indicazioni sulle caratteristiche esterne del prodotto finale individuando problemi che potrebbero ridurne la qualità.

Metriche per la qualità d'uso

Misurano il grado con cui il prodotto software permette agli utenti di svolgere le proprie attività con efficacia, produttività, soddisfazione e sicurezza.

B.2 Elenco delle metriche

B.2.1 Metriche per la qualità del processo

- MPR01: Soddifacimento requisiti obbligatori;
- MPR02: Indice di Gulpease;
- MPR03: Indice di correttezza ortografica;
- MPR04: Budget at Completion;
- MPR05: Actual Cost of Work Performed;
- MPR06: Budget Cost of Work Performed;
- MPR07: Budget Cost of Work Scheduled;
- MPR08: Schedule Variance;
- MPR09: Budget Variance;
- MPR10: Indice di risoluzione dei problemi.

B.2.2 Metriche per la qualità di prodotto

- MPDS01: Completezza dell'implementazione;
- MPDS02: Densità errori;
- MPDS03: Tempo di caricamento delle schermate;
- MPDS04: Facilità di utilizzo;
- MPDS05: Facilità di apprendimento;
- MPDS06: Profondità della gerarchia;
- MPDS07: Facilità di comprensione;
- MPDS08: Semplicità delle funzioni;
- MPDS09: Semplicità delle classi.