

NORME DI PROGETTO

Norme di Progetto v1.0.0

TheBlackCat

Responsabile	Stefano Scaglione
Redazione	Andrea Nalesso Stefano Scaglione
Verifica	Luca Allegro Davide Di Somma Giulia Albanello
Stato	Approvato
Uso	Interno
Destinato a	TheBlackCat prof. Tullio Vardanega prof. Riccardo Cardin
Email di contatto	theblackcat.swe@gmail.com



Diario delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
1.0.0	2018-04-12	Approvazione Documento	Stefano Scaglione	Responsabile
0.5.0	2018-03-20	Verificato sezione 4 Processi Organizzativi	Luca Allegro	Verificatore
0.4.1	2018-03-19	Modifica sottosottosezione 3.1.3 Strumenti in Processo di Supporto	Andrea Nalesso	Amministratore
0.4.0	2018-03-18	Verificato sezione 3 Processi di Supporto	Davide Di Somma	Verificatore
0.3.2	2018-03-17	Definito sezione 4 Processi Organizzativi	Andrea Nalesso	Amministratore
0.3.1	2018-03-16	Modifica sottosottosezione 3.1.3 Strumenti in Processo di Supporto	Andrea Nalesso	Amministratore
0.3.0	2018-03-15	Verificato sezione 3 Processi di Supporto	Davide Di Somma	Verificatore
0.2.1	2018-03-14	Definito sezione 3 Processi di Supporto	Andrea Nalesso	Amministratore
0.2.0	2018-03-13	Verificato sezione 2 Processi Primari	Giulia Albanello	Verificatore
0.1.1	2018-03-12	Incremento sottosezione 2.2 Processo di sviluppo in Processi Primari	Andrea Nalesso	Amministratore
0.1.0	2018-03-12	Verificato sezione 1 Introduzione	Giulia Albanello	Verificatore
0.0.3	2018-03-11	Definito sezione 2 Processi Primari	Andrea Nalesso	Amministratore



Versione	Data	Descrizione	Autore	Ruolo
0.0.2	2018-03-09	Definito sezione 1 Introduzione	Stefano Scaglione	Amministratore
0.0.1	2018-03-09	Creato scheletro del documento	Andrea Nalesso	Amministratore



Indice

1	Introduzione	7
1.1	Scopo del documento	7
1.2	Scopo del prodotto	7
1.3	Glossario	7
1.4	Riferimenti	7
1.4.1	Normativi	7
1.4.2	Informativi	7
2	Processi Primari	9
2.1	Fornitura	9
2.1.1	Scopo	9
2.1.2	Descrizione	9
2.2	Processo di sviluppo	9
2.2.1	Attività	9
2.2.1.1	Fase di Analisi	9
2.2.1.1.1	Studio di Fattibilità	9
2.2.1.1.2	Analisi dei Requisiti	10
2.2.1.2	Piano di Progetto	10
2.2.1.3	Piano di Qualifica	10
2.2.2	Norme	11
2.2.2.1	Classificazione dei Requisiti	11
2.2.2.2	Classificazione dei Casi d'uso	12
2.2.2.3	Codifica e convenzioni	13
2.2.2.3.1	Variabili, Funzioni e Commenti	13
2.2.2.3.2	Formattazione	13
2.2.3	Strumenti	14
2.2.3.1	SWEgo	14
3	Processi di Supporto	15
3.1	Processo di documentazione	15
3.1.1	Procedure	15
3.1.1.1	Modifica delle Norme	15
3.1.1.2	Formalizzazione di documento	15
3.1.2	Norme	15
3.1.2.1	Norme Tipografiche	15
3.1.2.1.1	Stile di testo	15
3.1.2.1.2	Segni di interpunzione	16
3.1.2.1.3	Composizione del testo	17
3.1.2.1.4	Formati ricorrenti	17



3.1.2.2	Componenti Grafiche	19
3.1.2.2.1	Immagini	19
3.1.2.2.2	Tabelle	19
3.1.2.3	Formattazione pagine	19
3.1.2.3.1	Frontespizio	19
3.1.2.3.2	Diario delle Modifiche	20
3.1.2.3.3	Indici	20
3.1.2.3.4	Altre pagine	20
3.1.2.4	Tipo di documento	20
3.1.2.4.1	Documenti preliminari	20
3.1.2.4.2	Documenti formali	21
3.1.2.4.3	Glossario	21
3.1.2.4.4	Verbalì	21
3.1.2.5	Versionamento dei Documenti	22
3.1.3	Strumenti	22
3.1.3.1	LaTeX	22
3.1.3.2	Editor LaTeX	23
3.1.3.3	GanttProject	23
3.1.3.4	PlantUML	23
3.1.3.5	StarUML	23
3.2	Processo di Verifica	23
3.2.1	Attività	24
3.2.1.1	Metriche	24
3.2.1.1.1	Metriche per i processi	24
3.2.1.1.2	Metriche per i documenti	26
3.2.1.1.3	Metriche per il software	27
3.2.1.2	Strumenti	31
3.2.1.3	Analisi	31
3.2.1.3.1	Analisi Statica	31
3.2.1.3.2	Analisi dinamica	32
3.2.1.4	Procedure	32
3.2.1.4.1	Gestione delle Anomalie	32
3.2.2	Strumenti per la verifica	33
3.3	Processo di Validazione	33
4	Processi Organizzativi	34
4.1	Processo di Gestione	34
4.1.1	Attività	34
4.1.1.1	Comunicazioni	34
4.1.1.1.1	Comunicazioni interne	34
4.1.1.1.2	Comunicazioni esterne	34



4.1.1.2	Riunioni	35
4.1.1.2.1	Riunioni interne	35
4.1.1.2.2	Riunioni esterne	35
4.1.1.3	Ticketing	35
4.1.1.4	Repository	36
4.1.2	Procedure	37
4.1.2.1	Apertura di una card	37
4.1.2.2	Assegnazione di una card	37
4.1.2.3	Modifica di una card	37
4.1.2.4	Rimozione di un card	38
4.1.2.5	Risoluzione di un card	38
4.1.3	Norme	38
4.1.3.1	Repository	38
4.1.3.1.1	Nome di File	38
4.1.3.1.2	Struttura del Repository	38
4.1.3.1.3	Commit	39
4.1.4	Strumenti	39
4.1.4.1	Sistema Operativo	39
4.1.4.2	Git	39
4.1.4.3	Github	40
4.1.4.4	Google Drive	40
4.1.4.5	Trello	40
4.1.4.6	Telegram	41
4.1.4.7	IFTTT	41
4.1.4.8	GroupAgree Bot	41

Elenco delle tabelle

1	Metriche per i processi	25
2	Metriche per i documenti	26
3	Metriche per il software	27



1 Introduzione

1.1 Scopo del documento

Questo documento definisce le norme interne che i membri di TheBlackCat dovranno seguire nello svolgimento del *capitolato_G* d'appalto Despeect proposto dall'azienda Mivoq SRL. Ogni membro del gruppo ha l'obbligo di visionare il documento e seguire rigorosamente le norme in esso contenute, al fine di garantire uniformità e coesione nel materiale prodotto. L'obiettivo del gruppo è quello di migliorare ed integrare nel tempo il documento, seguendo un'ottica incrementale.

1.2 Scopo del prodotto

Il prodotto ha lo scopo di fornire un'interfaccia grafica che consenta all'*utente_G* di operare agevolmente con le funzioni offerte dalla libreria *Speect_G*, ad esempio consentire l'esecuzione delle componenti di analisi e la visualizzazione del risultato mediante un grafo.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate del documento *Glossario v1.0.0*. Ogni occorrenza di vocaboli presenti nel *Glossario v1.0.0* è in corsivo e seguita dalla lettera 'g' maiuscola in pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0*;
- **Capitolato:** <http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C3.pdf> (ultima consultazione effettuata 2018-04-10).

1.4.2 Informativi

- **ISO-8601:** https://en.wikipedia.org/wiki/ISO_8601 (ultima consultazione effettuata 2018-04-10);
- **ISO 31-0:** https://en.wikipedia.org/wiki/ISO_31-0 (ultima consultazione effettuata 2018-03-15);



- **Documentazione Git:** <https://git-scm.com/doc> (ultima consultazione effettuata 2018-03-16);
- **GitHub:** <https://github.com/about> (ultima consultazione effettuata 2018-03-16)
- **Google Drive:** <https://support.google.com/drive> (ultima consultazione effettuata 2018-03-16)
- **Trello:** <https://trello.com/guide> (ultima consultazione effettuata 2018-04-01)
- **IFTTT:** <https://ifttt.com/about> (ultima consultazione effettuata 2018-04-03)
- **SWEgo:** <https://www.swego.it> (ultima consultazione effettuata 2018-04-03)
- **L^AT_EX:** <https://www.latex-project.org/about/> (ultima consultazione effettuata 2018-04-08)
- **GanttProject:** <http://www.ganttproject.biz/about> (ultima consultazione effettuata 2018-04-08)
- **PlantUML:** <http://plantuml.com/faq> (ultima consultazione effettuata 2018-04-10)
- **StarUML:** <http://staruml.io/> (ultima consultazione effettuata 2018-04-10)
- **Telegram:** <https://telegram.org/faq> (ultima consultazione effettuata 2018-04-10)



2 Processi Primari

2.1 Fornitura

2.1.1 Scopo

Questo *processo_G* ha lo scopo di trattare le norme e i termini che ogni componente del gruppo TheBlackCat si impegna a rispettare per diventare fornitori del proponente Mivoq SRL e dei *commitenti_G* prof. Tullio Vardanega e prof. Riccardo Cardin per quanto riguarda il prodotto Despeect.

2.1.2 Descrizione

Il gruppo si impegna a mantenere un costante dialogo con il referente dott. Giulio Paci dell'azienda Mivoq SRL, per avere un riscontro immediato ed efficace su ogni fase di avanzamento del progetto.

2.2 Processo di sviluppo

Di seguito sono descritti gli standard coi quali il gruppo deve affrontare le attività di progetto.

2.2.1 Attività

2.2.1.1 Fase di Analisi

2.2.1.1.1 Studio di Fattibilità

È compito del Responsabile di Progetto organizzare riunioni per permettere il confronto e lo scambio di opinioni tra i membri del team sui capitoli. Il primo documento per l'analisi dei requisiti è lo *Studio di Fattibilità v1.0.0* la cui realizzazione è affidata agli Analisti.

È compito del Responsabile di Progetto organizzare riunioni per permettere ai i membri del team di confrontarsi e scambiare le opinioni sui capitoli e di affidare agli Analisti la realizzazione dello *Studio di Fattibilità v1.0.0*. Gli Analisti, basandosi su quanto deciso nelle suddette riunioni, scrivono un'analisi che descriva i seguenti punti:

Dominio tecnologico e applicativo: si pensa a quanto sono attualmente conosciute dal team le tecnologie richieste.

Interesse strategico: si valuta l'interesse strategico del team, in relazione al *capitolato_G* in analisi.



Individuazione dei rischi: si comprendono quali sono i punti critici della realizzazione, individuando eventuali rischi e le problematiche che possono insorgere in corso d'opera.

2.2.1.1.2 Analisi dei Requisiti

Dopo aver completato lo *studio di fattibilità_G* è compito degli Analisti redigere il documento Analisi dei Requisiti. Scopo di tale documento è quello di elencare in maniera non ambigua e sintetica tutti i requisiti recuperabili da più fonti, nel dettaglio:

- *capitolato_G* d'appalto;
- verbali di riunioni interne o esterne;
- casi d'uso.

2.2.1.2 Piano di Progetto

Il Responsabile, aiutato dagli Amministratori, deve redigere un piano da seguire nella realizzazione del progetto. Il documento deve contenere:

Analisi dei rischi: si identificano e si analizzano i rischi che potrebbero insorgere nel corso del progetto, capendo con che probabilità si possono verificare, loro livello di gravità e che impatti possono avere sull'esecuzione del progetto. Per ogni rischio si deve individuare un modo per affrontarlo.

Pianificazione: si pianificano le attività da svolgere nel corso del progetto, ponendo scadenze temporali ben precise.

Preventivo e Consultivo: sulla base della pianificazione si stima la quantità di lavoro necessaria per ogni fase, così da realizzare un *preventivo_G* sul costo totale del progetto. Alla fine di ogni attività si redige un consultivo del periodo per tracciare l'andamento rispetto a quanto si era preventivato.

2.2.1.3 Piano di Qualifica

I Verificatori devono scegliere una strategia da adottare per la verifica e la validazione del lavoro prodotto. Il documento deve contenere:

Visione generale: si stabiliscono degli standard per verificare la qualità dei processi e del prodotto, tenendo in considerazione il tempo e le risorse a disposizione. Vengono classificate le responsabilità e l'ordine delle attività da sottoporre a verifica.



Obiettivi di qualità: si stabiliscono gli obiettivi da raggiungere in merito alla qualità dei processi e del prodotto, stabilendo delle metriche.

Specifica dei test: si stabiliscono dei test da eseguire sul prodotto.

Resoconto delle attività di verifica: alla fine di ogni attività si devono riportare le metriche calcolate e un resoconto sulla verifica.

Esiti delle revisioni: ad ogni revisione si devono riportare le modifiche apportate in base alle segnalazioni del *committente_G*.

2.2.2 Norme

2.2.2.1 Classificazione dei Requisiti

Ogni *requisito_G* individuato durante l'Analisi deve necessariamente, per motivi di tracciabilità, essere univocamente identificato con una stringa nel formato:

$$R[\text{importanza}][\text{tipo}][\text{codice}]$$

dove:

Importanza: può assumere i seguenti valori:

- 0: *requisito_G* obbligatorio;
- 1: *requisito_G* desiderabile;
- 2: *requisito_G* opzionale.

Tipo: può assumere i seguenti valori:

- F: *requisito_G* funzionale;
- Q: *requisito_G* qualitativo;
- P: *requisito_G* prestazionale;
- V: *requisito_G* vincolante.

Codice: è il codice numerico gerarchico univoco (p.es. 4.2).

Per ogni *requisito_G* si specifica inoltre:

Descrizione: sintetica e non ambigua.

Fonti: possono essere uno o più elementi tra:

- *capitolato_G* d'appalto;



- verbali di riunioni interne;
- verbali di riunioni esterne.

Deve essere presente una tabella che consenta il tracciamento tra i requisiti e i casi d'uso loro associati.

2.2.2.2 Classificazione dei Casi d'uso

Ogni caso d'uso è identificato da un codice, che segue il seguente formalismo:

UC[codice_padre].[codiceLivello]

Dove:

codice_padre: numero che identifica univocamente i casi d'uso.

codiceLivello: numero progressivo che identifica i sottocasi e può a sua volta includere altri livelli.

Per ogni caso d'uso sono inoltre specificati:

Titolo: un breve titolo riassuntivo.

Attori: tutti gli attori coinvolti nel caso d'uso.

Descrizione: una descrizione sintetica e completa del caso d'uso.

Precondizione: le condizioni sempre vere all'origine del caso d'uso e necessarie per la sua esecuzione.

Postcondizione: le condizioni sempre vere al termine del caso d'uso.

Scenario principale: descrive il flusso di operazioni di cui è composto il caso d'uso.

Scenari alternativi: descrive situazioni anomale che possono verificarsi durante l'esecuzione e come vengono gestite.

Diagramma UML: diagramma rappresentante il caso d'uso se significativo.



2.2.2.3 Codifica e convenzioni

Al fine di massimizzare l'efficacia e l'efficienza nello sviluppo del prodotto, è necessario definire convenzioni e norme da rispettare durante ogni fase di codifica. Il rispetto e la standardizzazione delle norme di codifica comporta i seguenti vantaggi:

- massima comprensibilità;
- massima manutenibilità.

Sono stati considerati i seguenti aspetti per la definizione di norme di codifica della *team di sviluppo_G*:

- struttura del codice;
- semantica del codice.

Entrambi i precedenti aspetti si riflettono sui seguenti elementi del codice

- nomenclatura variabili e funzioni, stesura commenti;
- formattazione(spaziature, tabulazioni etc).

2.2.2.3.1 Variabili, Funzioni e Commenti

- i commenti al codice devono essere esplicativi;
- i commenti al codice devono essere in lingua inglese;
- i nomi delle funzioni devono essere esplicativi;
- i nomi delle funzioni devono essere in lingua inglese;
- i nomi delle variabili devono essere esplicativi;
- i nomi delle variabili devono essere in lingua inglese.

2.2.2.3.2 Formattazione

- rispettare al massimo lo stile di formattazione scelto;
- strutturare codice in modo che sia massimamente leggibile;
- usare spaziature tra gli elementi del linguaggio se possibile.



The screenshot shows the SWEgo web application. At the top is a blue header with the SWEgo logo on the left and a navigation menu on the right: FONTE, ATTORI, REQUISITI, USE CASE, TRACCIAMENTO (highlighted), PROFILO, CONTATTI, DONAZIONI, F.A.Q. Below the header, a message reads: "Visualizza il tracciamento tra Use Case e Requisiti. Puoi [tracciare](#) altri Use Case con i Requisiti." The main content is a table with four columns: "Codice Use Case", "Nome Use Case", "Codice Requisiti", and "Operazioni". The table contains 11 rows of data. Each row in the "Operazioni" column has two icons: a pencil (edit) and a trash can (delete).

Codice Use Case	Nome Use Case	Codice Requisiti	Operazioni
UC1	Chiusura del programma	R0F2	
UC2	Configurazione di Speect	R0F3	
UC9J	Esportazione della configurazione	R1F6 R1F6.1 R1F6.2 R1F6.3	
UC42	Rimozione componente da un utterance-type	R2F9J	
UC43	Cambio posizione componenti in utterance-type	R2F9.2	
UC5J	Esecuzione utterance-type	R0F10	
UC5.2	Esecuzione componente	R1F1	
UC5.3	Esecuzione componente in utterance-processors	R2F14	
UC5.4	Esecuzione componente in feature-processors	R2F15	
UC4IJ	Aggiunta utterance-processor	R2F16	
UC4I2	Aggiunta feature-processor	R2F17	

Figura 1: Screenshot di SWEgo

2.2.3 Strumenti

2.2.3.1 SWEgo

Si è scelto di adottare il software *SWEgo_G* in quanto permette di:

- tracciare gli Use Case con i requisiti;
- visualizzare i grafici relativi alla copertura dei requisiti;
- generare automaticamente i diagrammi dei casi d'uso con *PlantUML_G*.

Il software appoggia su un database MySQL ed è reperibile all'indirizzo <http://www.swego.it>.



3 Processi di Supporto

3.1 Processo di documentazione

Nelle sezioni seguenti è spiegato il modello con il quale la documentazione viene prodotta, incrementata e formalizzata durante l'intero *ciclo di vita del software_G*.

3.1.1 Procedure

3.1.1.1 Modifica delle Norme

Variazioni al documento possono essere proposte da parte dei componenti del gruppo al Responsabile di Progetto, il quale può decidere se accettare la modifica, rifiutarla o porre la questione agli altri membri del gruppo: l'ultima decisione spetta comunque al Responsabile di Progetto. La discussione può essere formale (ovvero avvenire all'interno di una riunione) oppure informale (utilizzando lo strumento di comunicazione interna): è a discrezione del Responsabile di Progetto decidere se avvalersi dell'una, dell'altra o di entrambe. Una volta approvata una modifica, il Responsabile di Progetto è tenuto a informare i membri del gruppo del cambiamento mediante lo strumento di comunicazione interna e ad assegnare a un componente del gruppo, usando il sistema di *ticketing_G*, il compito di correggere il documento.

3.1.1.2 Formalizzazione di documento

Ogni documento non formale di cui sia stata completata la stesura dovrà essere sottoposto al Responsabile di Progetto, che a sua volta si occuperà di incaricare i Verificatori di controllarne il contenuto e la forma. Un documento può essere formalizzato e approvato solo sotto consenso del Responsabile di Progetto. Solo se un documento è formalizzato può essere distribuito pubblicamente.

3.1.2 Norme

3.1.2.1 Norme Tipografiche

3.1.2.1.1 Stile di testo

Grassetto: questo stile si applica ai titoli, agli elementi di un elenco che riassumono il contenuto del paragrafo e alle parole che costituiscono il punto focale dell'argomento e dunque devono essere messe in risalto.

Corsivo: questo stile si applica per indicare percorsi di cartelle, file e nomi dei documenti, per riportare passaggi provenienti da fonti esterne, per gli elementi di un elenco di secondo livello e i termini del glossario.



Sottolineato: questo stile verrà usato per far risaltare l'importanza della parola nella frase. Ad esempio, verrà usato con le particelle di negazione o con parole che stanno ad indicare l'obbligatorietà dell'azione.

Monospaziato: si applica un *font monospaziato_G* per aumentare la leggibilità del codice, dei comandi, dei percorsi e degli indirizzi web.

Glossario: questo stile si applica esclusivamente per segnalare che il termine ha una corrispondenza nel glossario: è caratterizzato dalla marcatura della parola in stile corsivo e, per ogni occorrenza del termine all'interno del documento, da una 'g' corsiva posta a pedice.

Maiuscolo: l'utilizzo di parole le cui lettere sono tutte maiuscole deve limitarsi solo agli acronimi. Le lettere iniziali maiuscole si usano dopo segni di punteggiatura quali il punto fermo, il punto di domanda e il punto esclamativo. Vengono inoltre usate per indicare i nomi propri, i nomi dei ruoli di progetto che fanno riferimento a persone specifiche (p.es. *Analista_G*). I nomi delle fasi e attività di progetto presentano maiuscola l'iniziale di ogni parola.

3.1.2.1.2 Segni di interpunzione

Punteggiatura: deve essere utilizzata in modo attento per raggruppare parti di frase dal senso coeso e coerente. L'utilizzo del punto è richiesto per terminare un concetto e passare alla discussione di un altro argomento affine.

Spazi: nessun segno di punteggiatura, eccetto le parentesi aperte, è anticipato da spazi. La virgola deve essere seguita da uno spazio.

Parentesi: all'interno di una frase possono essere presenti univocamente le parentesi tonde. Possono essere impiegate ogni qual volta si vuole fornire un sinonimo del termine, oppure un esempio.

Puntini di sospensione: da limitare il più possibile.

Virgolette singole: si usano per citare un singolo carattere (p.es. 'g').

Virgolette doppie: si usano per le citazioni.

Slash: il simbolo '/' si usa per indicare l'esclusione o l'entrambe (p.es. il presidente e/o il segretario) oppure nei percorsi di cartelle, di indirizzi web e nel codice.



3.1.2.1.3 Composizione del testo

Elenchi: ogni voce di un elenco semplice (i cui elementi sono costituiti da un solo enunciato), se preceduto dal segno di due punti, comincia con l'iniziale minuscola e termina con il punto e virgola tranne l'ultima, seguita dal punto fermo; altrimenti ogni voce comincia con l'iniziale maiuscola e termina con il punto fermo. Ogni voce di un elenco complesso (in cui almeno uno degli elementi sia composto da più di un enunciato) comincia con l'iniziale maiuscola (anche dopo il segno di due punti) e termina con il punto fermo.

Paragrafi: i paragrafi hanno tutti lo stesso livello di indentazione, devono essere giustificati e sono separati da una riga vuota.

Esempi: quando viene indicato un esempio, lo si deve fornire in linea con il testo, racchiuso tra parentesi tonde e usando il seguente formalismo:

(p.es. ...)

3.1.2.1.4 Formati ricorrenti

Numeri: Lo stile dei numeri deve aderire allo standard *SI/ISO 31-0_G* (p.es. 4 200,42).

Denaro: Qualora sia necessario specificare una somma di denaro si deve far seguire al numero il simbolo della valuta separato da uno spazio (p.es. 5 489,90 €).

Date: Lo stile delle date deve aderire allo standard *ISO 8601_G* con il formato seguente:

AAAA-MM-GG

dove:

- AAAA è l'anno in quattro cifre;
- MM è il mese in due cifre;
- GG è il giorno in due cifre.

Orari: Lo stile degli orari deve aderire allo standard *ISO 8601_G* con il formato seguente:

HH:MM

dove:



- HH è l'ora in due cifre;
- MM sono i minuti in due cifre.

Luoghi: Lo stile dell'indicazione di un luogo deve aderire al formato:
Città (PR) in via Nome della Via, #.

Nomi di ruoli I nomi dei ruoli di progetto che fanno riferimento a persone specifiche presentano l'iniziale maiuscola (p.es. *Analista_G*).

Nomi di documenti: Ogni iniziale di parola che non sia una preposizione va con la lettera iniziale maiuscola, inoltre, se è inclusa la versione del documento, l'intero nome è in corsivo (p.es. *Analisi dei Requisiti v1.0.0*). Esiste un apposito comando `\LTeX` per ogni documento.

Nomi di file: I nomi di file sono indicati in corsivo comprensivi di estensione (p.es. *NormeDiProgetto.pdf*). Esiste un apposito comando `\LTeX` per ogni file.

Nomi propri di persona: Il nome precede il cognome. Se la persona possiede un titolo noto che è necessario specificare esso va fatto precedere al nome con notazione minuscola e abbreviata (p.es. dott. Mario Rossi).

Sigle: Le sigle dei documenti e delle revisioni potranno essere utilizzate solo nelle tabelle per questioni di spazio. In particolare sono previste le seguenti sigle:

- NP: Norme di Progetto;
- Gl: Glossario;
- SF: Studio di Fattibilità;
- AR: Analisi dei Requisiti;
- PQ: Piano di Qualifica;
- PP: Piano di Progetto;
- ST: Specifica Tecnica;
- DP: Definizione di Prodotto;
- MU: Manuale Utente;
- RR: Revisione dei Requisiti;
- RP: Revisione di Progettazione;
- RQ: Revisione di Qualifica;
- RA: Revisione di Accettazione;
- VI: Verbale Interno;
- VE: Verbale Esterno.



3.1.2.2 Componenti Grafiche

3.1.2.2.1 Immagini

Per garantire il massimo della resa grafica, le immagini da inserire all'interno dei documenti dovranno, ove possibile, essere in formato vettoriale (PDF_G o, in alternativa, generate dai pacchetti $TikZ_G$ e PGF_G). Ogni immagine deve essere corredata di numero progressivo e didascalia, entrambi riportati nell'indice delle figure.

3.1.2.2.2 Tabelle

Nella stesura delle tabelle sarà obbligatoria l'osservazione delle regole di seguito descritte, aumentandone la tracciabilità e la facilità di lettura.

Didascalia: ogni tabella dovrà essere descritta da un numero progressivo e una didascalia riassuntiva i contenuti della tabella.

Intestazioni colonne: l'intestazione conterrà i titoli delle colonne. Esse avranno bianco come colore di sfondo e testo nero. Fa eccezione il diario, il cui colore dello sfondo dell'intestazione è carminio (■) e colore dello sfondo delle righe interne è □ (bianco) alternato a ■ (grigio).

Filetti: per ragioni tipografiche, le tabelle non devono presentare filetti verticali o filetti doppi e fare un uso limitato dei filetti orizzontali.

3.1.2.3 Formattazione pagine

3.1.2.3.1 Frontespizio

Sulla prima pagina del documento sono visibili centrati e in ordine dall'alto verso il basso:

- nome del progetto;
- il logo del team;
- titolo del documento comprensivo di versione;
- il nome del team;
- tabella informativa del documento contenente:
 - nome e cognome dei Responsabili;
 - nome e cognome dei Redattori;
 - nome e cognome dei Verificatori;



- stato del documento;
- uso (interno o esterno);
- lista di distribuzione;
- email di contatto.

3.1.2.3.2 Diario delle Modifiche

Dalla seconda pagina di ogni documento è visibile il diario delle modifiche. Tale tabella è strutturata in 5 colonne:

Versione: definisce la versione del documento.

Data: definisce la data in cui il documento ha subito le modifiche descritte in Descrizione.

Descrizione: descrizione sintetica delle modifiche effettuate.

Autore: autore delle modifiche.

Ruolo: ruolo dell'autore delle modifiche.

La tabella è ordinata per data in ordine decrescente. In questo modo l'ultimo cambiamento effettuato sarà sempre presente nella prima riga della tabella.

3.1.2.3.3 Indici

In ogni documento, dopo il diario delle modifiche, è presente un indice rispettivamente per le sezioni, le figure e le tabelle. La numerazione di ogni indice deve partire da 1.

3.1.2.3.4 Altre pagine

Nella testatina di ogni pagina è presente a sinistra il logo del gruppo e a destra il numero e il nome della sezione. A piè di ogni pagina è presente a sinistra il nome e la versione del documento, mentre a destra il numero della pagina del documento e il numero totale di pagine.

3.1.2.4 Tipo di documento

3.1.2.4.1 Documenti preliminari

Fino all'approvazione finale da parte del Responsabile di Progetto ogni documento è da considerarsi parziale e passibile di modifica secondo le modalità specificate.



3.1.2.4.2 Documenti formali

Un documento formale è da considerarsi tale se e soltanto se è stato approvato dal *Responsabile di Progetto_G*. Una volta approvato, tale documento è considerato disponibile per ogni membro della lista presente all'interno della lista di distribuzione.

3.1.2.4.3 Glossario

Per evitare ambiguità nel linguaggio e massimizzare la comprensione dei documenti, è definito un glossario. Ogni occorrenza, all'interno di qualsiasi documento, dei vocaboli presenti nel Glossario è posta in corsivo e marcata da una 'g' maiuscola a pedice (p.es. *Voce di glossario_G*).

3.1.2.4.4 Verbali

Dopo ogni riunione interna o esterna spetta al segretario verbalizzante redigere il documento. I verbali non sono soggetti a *versionamento_G*. Il codice identificativo di ogni *verbale_G* segue la seguente struttura:

VX[data]

dove:

X: identifica uno dei seguenti tipi di *verbale_G*:

- E: *verbale_G* esterno;
- I: *verbale_G* interno.

data: è la data della riunione alla quale il *verbale_G* fa riferimento nel formato AAAA-MM-GG.

Ogni *verbale_G* deve presentare le seguenti informazioni, rispettando l'ordine in cui sono elencate:

- data, ora e luogo della riunione;
- i componenti del gruppo presenti;
- eventuali persone presenti esterne al gruppo;
- il segretario verbalizzante;
- l'ordine del giorno;



- Le decisioni assunte: ogni decisione deve essere identificata da un codice identificativo che segue la seguente struttura:

[nome]_[numero]

dove:

Nome: nome completo del *verbale_G*.

Numero: numero della decisione presa a partire da 1.

- gli eventuali problemi irrisolti.

3.1.2.5 Versionamento dei Documenti

Ogni documento presenta un numero di versione dopo il nome nel formato seguente:

vX.Y.Z

dove:

- X: è un numero intero, che parte da 0, incrementato di una unità in occasione di una formalizzazione da parte del Responsabile di Progetto, l'incremento di X riporta Y e Z a 0.
- Y: è un numero intero, che parte da 0, incrementato ad ogni verifica effettuata sul documento da parte del Verificatore. L'incremento di Y comporta il ritorno a 0 di Z.
- Z: è un numero intero, che parte da 0, incrementato ad ogni modifica significativa del documento.

3.1.3 Strumenti

3.1.3.1 LaTeX

Come linguaggio di markup necessario alla stesura di documentazione necessaria per il progetto è stato scelto \LaTeX per le seguenti motivazioni:

- è gratuito;
- è *open-source_G*;
- i documenti in \LaTeX sono versionabili;
- esistono diversi motori di composizione tipografica, quindi non si è legati a un *vendor_G* particolare;



- disponibile della ottima documentazione, anche gratuita e in italiano;
- ottimo il supporto online da parte della community di utilizzatori di \LaTeX ;
- notevole disponibilità pacchetti contenenti estensioni al linguaggio.

3.1.3.2 Editor \LaTeX

Si lascia piena libertà di scelta relativamente all'editor \LaTeX scelto: è richiesto però l'uso di UTF-8 senza BOM_G come encoding dei file dei documenti.

3.1.3.3 GanttProject

Per la creazione dei diagrammi di $Gantt_G$ presenti nel documento *Piano di Progetto v1.0.0* è stato scelto $GanttProject_G$ ($\geq 2.8.0$).

3.1.3.4 PlantUML

I diagrammi dei casi d'uso vengono automaticamente generati da $SWEgo_G$, il quale produce una descrizione testuale del diagramma che può essere successivamente utilizzata da $PlantUML_G$ per produrre i diagrammi in formato visuale (p.es. SVG_G , png_G , \LaTeX).

3.1.3.5 StarUML

Per i restanti diagrammi UML_G si è scelto di utilizzare $StarUML_G$. I motivi di questa scelta sono:

- semplicità d'uso (rispetto a $papyrus_G$);
- possibilità di esportare in formato SVG_G ;
- ridotto uso delle risorse del computer.

3.2 Processo di Verifica

Il $processo_G$ di verifica si occupa di evitare che vengano introdotti errori nel prodotto a seguito dell'esecuzione delle attività dei processi svolti nella fase in esame. Il $processo_G$ è suddiviso in due attività:

Analisi: consiste nell'analisi del codice sorgente e la sua successiva esecuzione. Viene effettuata tramite due tecniche, l'analisi statica e l'analisi dinamica.

Test: definisce tutti i test che vengono eseguiti sul prodotto software.



3.2.1 Attività

3.2.1.1 Metriche

Per garantire la qualità del lavoro del gruppo, i Verificatori hanno definito delle metriche, riportandone gli obiettivi di qualità nel *Piano di Qualifica v1.0.0*. Tali metriche devono rispettare la seguente notazione:

$$M[X][Y]$$

dove:

X: indica se la *metrica_G* si riferisce a processi, prodotto documento o prodotto software e può assumere i seguenti valori:

- PC: per indicare una *metrica_G* per il *processo_G*;
- PD: per indicare una *metrica_G* per il documento;
- PS: per indicare una *metrica_G* per il software.

Y: indica il codice univoco della *metrica_G* (numero intero incrementale a partire da 1).

3.2.1.1.1 Metriche per i processi

Le seguenti metriche rappresentano un indicatore utilizzato per monitorare e prevedere l'andamento delle principali variabili critiche del progetto: tempi e costi. Sono utilizzate metriche di tipo *consuntivo_G* perché danno un riscontro immediato sullo stato attuale del progetto. Ad ogni incremento verranno valutati tali indici e, se necessario, verranno stabiliti opportuni provvedimenti da parte del Responsabile di Progetto.



Nome	Codice	Descrizione
Schedule Variance (SV)	MPC1	<p>È un indicatore di efficacia.</p> $SV = BCWP - BCWS$ <p>Dove:</p> <p>BCWP: indica il valore delle attività realizzate alla data corrente;</p> <p>BCWS: indica il costo pianificato per realizzare le attività di progetto alla data corrente.</p> <p>Se $SV > 0$ indica che il lavoro viene prodotto in anticipo rispetto quanto pianificato, viceversa mentre se negativo.</p>
Budget Variance (BV)	MPC2	<p>Indica se alla data corrente si è speso di più o di meno rispetto a quanto pianificato. È un indicatore che ha un valore contabile e finanziario.</p> $BV = BCWS - ACWP$ <p>Dove:</p> <p>BCWP: indica il valore delle attività realizzate alla data corrente;</p> <p>BCWS: indica il costo pianificato per realizzare le attività di progetto alla data corrente.</p> <p>Se $BV > 0$ significa che l'attuazione del progetto sta consumando il proprio budget con minor velocità rispetto a quanto pianificato, viceversa se negativo.</p>
Rischi non preventivati	MPC3	<p>Indicatore che evidenzia i rischi non preventivati. La misurazione viene effettuata attraverso un indice numerico che viene incrementato nel momento in cui si manifesta un rischio non individuato nell'attività di <i>analisi dei rischi</i>.</p>

Tabella 1: Metriche per i processi



Nome	Codice	Descrizione
Gulpease	MPD1	<p>L'indice $Gulpease_G$ è un indice di leggibilità di un testo tarato sulla lingua italiana. Rispetto ad altri ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. Permette di misurare la complessità dello stile di un documento.</p> <p>L'indice $Gulpease_G$ considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere. La formula per il suo calcolo è:</p> $89 + \frac{300 \times (\text{numero delle frasi}) - 10 \times (\text{numero delle lettere})}{(\text{numero delle parole})}$ <p>Il risultato è un valore compreso nell'intervallo tra 0 e 100, dove il valore 100 indica la più alta leggibilità. Un indice inferiore a 80 indica documenti di difficile leggibilità per chi ha la licenza elementare, inferiore a 60 per chi ha la licenza media, inferiore a 40 per chi ha un diploma superiore.</p>

Tabella 2: Metriche per i documenti

3.2.1.1.2 Metriche per i documenti

Come $metrica_G$ per i documenti redatti si è scelto di utilizzare l'indice di leggibilità. Vi sono a disposizione molti indici di leggibilità, ma i più importanti sono per la lingua inglese. Si è deciso quindi di adottare un indice di leggibilità per la lingua italiana.



3.2.1.1.3 Metriche per il software

Tabella 3: Metriche per il software

Nome	Codice	Descrizione
Copertura requisiti obbligatori	MPS1	<p>Questa <i>metrica_G</i> permette di tracciare il numero di requisiti obbligatori coperti dal prodotto software. Viene calcolata in forma percentuale utilizzando la seguente formula:</p> $\frac{(\# \text{ requisiti obbligatori soddisfatti})}{(\# \text{ totale requisiti obbligatori})}$
Copertura requisiti desiderabili	MPS2	<p>Questa <i>metrica_G</i> permette di tracciare il numero di requisiti desiderabili coperti dal prodotto software. Viene calcolata in forma percentuale utilizzando la seguente formula: ne calcolata in forma percentuale utilizzando la seguente formula:</p> $\frac{(\# \text{ requisiti desiderabili soddisfatti})}{(\# \text{ totale requisiti desiderabili})}$
Densità di failure	MPS3	<p>Indica la percentuale di operazioni di testing che si sono concluse in failure. La formula è:</p> $F = \frac{FR}{TE} \times 100$ <p>dove FR è il numero di failure rilevati durante l'attività di testing e TE è il numero di test-case eseguiti.</p>

Continua nella prossima pagina



Continua dalla pagina precedente

Nome		Codice	Descrizione
Capacità di analisi di failure		MPS4	Indica la percentuale di failure registrate, delle quali sono state individuate le cause. La misurazione si effettua attraverso la formula $I = \frac{FI}{FR} \times 100$ dove FI è il numero di failure delle quali sono state individuate le cause e FR è il numero di failure rilevate.
Sistemi operativi supportati		MPS5	Indica su quanti sistemi operativi diversi può essere utilizzato il software. La misurazione viene eseguita mediante un indice numerico che indica il numero dei sistemi operativi sui quale è supportato.
Numero di metodi per classe		MPS6	Indica il numero di metodi definiti in una <i>classe_C</i> ; un valore molto alto potrebbe indicare una cattiva decomposizione delle funzionalità a livello di progettazione. La misurazione viene eseguita mediante un indice numerico che indica il numero di metodi definiti in una classe.
Numero di parametri per metodo		MPS7	Indica il numero di parametri passati ad un metodo; un valore molto alto potrebbe indicare un metodo troppo complesso e non efficacemente suddiviso in sotto-metodi. La misurazione viene effettuata mediante un indice numerico che indica il numero di parametri passato ad un metodo.

Continua nella prossima pagina



Continua dalla pagina precedente

Nome	Codice	Descrizione
Complessità ciclomatica	MPS8	La complessità ciclomatica è una $metrica_G$ software che indica la complessità di un programma misurando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso. Valori troppo elevati sono indice di una scarsa manutenibilità del codice, mentre valori troppo bassi potrebbero indicare scarsa efficienza. La misurazione viene eseguita tramite indice numerico che indica il numero cammini percorribili nel grafo di controllo di flusso di un metodo.
Numero di livelli di annidamento	MPS9	Indica il numero di funzioni o procedure chiamate all'interno di un metodo; un valore elevato di tale indice implica un'alta complessità ed un basso livello di astrazione del codice. La misurazione viene effettuata attraverso un indice numerico che indica il numero di chiamate a funzioni o procedure presenti all'interno di un metodo.

Continua nella prossima pagina



Continua dalla pagina precedente

Nome	Codice	Descrizione
Linee di codice per linee di commento	MPS10	Indica la percentuale di linee di commento presenti all'interno del codice sorgente; la loro presenza permette una più semplice comprensione ed un maggior livello di manutenibilità di quanto prodotto. Un valore elevato di tale <i>metrica_G</i> può indicare un'eccessiva complessità del metodo. Si misura in percentuale $P = \frac{C}{SLOC} \times 100$ dove C è il numero di linee di commento presenti nel codice e SLOC è il numero di linee di codice prodotte;
Copertura del codice	MPS11	Indica in percentuale di codice che è stato testato con esito positivo tramite i test eseguiti. Viene calcolata tramite la seguente formula: $\frac{\# \text{ test passati}}{\# \text{ totale test richiesti}}$ Un valore elevato di tale <i>metrica_G</i> diminuisce la probabilità di errori, andando a garantire un livello maggiore di affidabilità del software.

Si conclude dalla pagina precedente



3.2.1.2 Strumenti

È stato inoltre utilizzato per calcolare l'indice Gulpease uno strumento disponibile al seguente indirizzo: http://farfalla-project.org/readability_static/ (ultima consultazione effettuata in data 2018-04-12)

3.2.1.3 Analisi

3.2.1.3.1 Analisi Statica

L'analisi statica è un *processo_G* di verifica che permette di rilevare le anomalie all'interno dei file di codice e documentazione. Si può svolgere con l'ausilio di due tecniche:

Walkthrough: Tipologia di analisi nella quale il documento o codice da analizzare è percorso nella sua interezza. Questa tipologia di analisi ha i seguenti vantaggi:

- è semplice da eseguire;
- non richiede la definizione di liste di controllo.

E i seguenti svantaggi:

- onerosa in termini temporali e meno focalizzata rispetto ad altri metodi di analisi (p.es. inspection).

L'analisi tramite *Walkthrough_G* è particolarmente indicata nella fasi iniziali di stesura di un documento dove le criticità comuni sono ancora sconosciute.

Inspection: Tipologia di analisi nella quale ogni *verificatore_G* è dotato di una "lista di controllo" contenente quali criticità ed errori potrebbero essere presenti nella documentazione o codice da verificare. Questa tipologia di analisi ha i seguenti vantaggi:

- permette un'analisi molto mirata (maggiore rispetto a *Walkthrough_G*).

e i seguenti svantaggi:

- richiede la definizione di una o più "liste di controllo".

In sintesi, l'analisi tramite inspection è particolarmente indicata nella fasi successive e finali di stesura di un documento dove è possibile definire criticità comuni e "fattorizzabili".



3.2.1.3.2 Analisi dinamica

L'analisi dinamica è applicabile solo a componenti software e viene svolta tramite l'esecuzione di test su essi. Essi ne verificheranno il funzionamento e, in caso di anomalie, aiutano ad identificarle. Per garantire risultati attendibili è necessario che i test siano ripetibili. Quindi dato lo stesso input, lo stesso test deve produrre lo stesso output se eseguito nello stesso ambiente. Solo un test con queste caratteristiche è in grado di riscontrare problemi e verificare la correttezza del prodotto software. Per ogni test deve essere definito:

ambiente: consiste nel sistema hardware e software sui quali è pianificata l'esecuzione del test sul prodotto. Va inoltre specificato uno stato iniziale dal quale il test deve partire.

specifica: consiste nella specifica degli input e degli output attesi.

procedure: consiste nella specifica di ulteriori istruzioni su come va eseguito il test e su come vanno interpretati i risultati.

3.2.1.4 Procedure

3.2.1.4.1 Gestione delle Anomalie

Una volta che l'assegnatario ritiene di aver concluso il compito assegnato, il Verificatore ha tre compiti:

1. verificare l'effettivo completamento del compito assegnato;
2. verificare la correttezza di quanto prodotto;
3. comunicare al Responsabile la terminazione della verifica.

Se il Verificatore ritiene che il compito sia stato completato in modo adeguato, allora il Responsabile può risolvere la $card_G$ assegnata. Una volta verificato l'effettivo completamento del compito assegnato, il Verificatore procede a definire l'insieme di anomalie reperite in quanto prodotto dal redattore. Ogni anomalia riscontrata all'interno di quanto prodotto è inserita in una lista. Ogni anomalia è definita da:

- nome del file;
- posizione;
- tipo anomalia.

Tale lista è consegnata al Responsabile, il quale deve procedere a definire le $card_G$ necessarie alla risoluzione di ogni anomalia. Inoltre, la lista di anomalie riscontrate dal Verificatore definisce una "lista di controllo" utile ad ogni verifica successiva di quanto prodotto.



3.2.2 Strumenti per la verifica

3.3 Processo di Validazione

Il *processo_G* di *validazione_G* consente di accertare se il prodotto software finale è conforme alle attese, quindi efficace. Questo *processo_G* consiste in due attività principali:

- test di sistema e di *validazione_G*;
- *collaudo_G*.

I test di sistema sono eseguiti tramite test di integrazione che non verificano la correttezza logica del software scritto, bensì la conformità rispetto alle attese delle funzionalità presenti. Vista la natura del prodotto, che presenta un'importante componente di interfaccia utente, i test di *validazione_G* verranno eseguiti su di essa.

Le attività del *processo_G* di *validazione_G* saranno le seguenti:

- pianificazione dei test da eseguire con relativo tracciamento;
- conduzione dei test in modo tale da forzare l'applicazione nei punti critici e dove è più probabile il verificarsi di errori;
- verifica del soddisfacimento dei requisiti seguendo le informazioni di tracciamento.

Sarà compito dei progettisti definire la pianificazione e la progettazione dei test, compito dei Verificatori eseguirli e tracciarne i risultati tramite gli strumenti preposti. Al fine di garantire la necessaria obiettività e imparzialità nell'esecuzione, chi eseguirà un determinato test dovrà per forza essere una persona distinta da chi l'ha progettato ed implementato.



4 Processi Organizzativi

4.1 Processo di Gestione

4.1.1 Attività

4.1.1.1 Comunicazioni

4.1.1.1.1 Comunicazioni interne

Le comunicazioni interne al team avvengono attraverso l'applicativo *Telegram_G*.

È possibile al suo interno definire con un tag argomenti rilevanti. È definita inoltre una *applet_G* (tramite il bot *IFTTT_G*) che inoltra nel gruppo ufficiale le e-mail ricevute presso la casella di posta ufficiale.

4.1.1.1.2 Comunicazioni esterne

Per le comunicazioni esterne è stata creata la casella di posta elettronica:

theblackcat.swe@gmail.com

Questa casella è l'unico canale di comunicazione ufficiale verso l'esterno del team. Solo il Responsabile di Progetto è autorizzato all'invio di messaggi tramite la suddetta casella di posta elettronica. Ogni comunicazione inviata alla casella di posta elettronica ufficiale è visualizzata all'interno del gruppo *Telegram_G*. Di seguito è spiegato il contenuto e la forma che ogni e-mail deve avere.

Oggetto: Deve essere chiaro e devono essere evitate il più possibile ambiguità e somiglianza con tutti gli altri messaggi presenti nella casella di posta. Qualora la comunicazione sia inviata al proponente (Mivoq SRL) l'oggetto deve contenere la dicitura [UNIPD-TTS].

Formula d'apertura: La formula d'apertura deve cominciare con un aggettivo di cortesia o un saluto e il nome del destinatario preceduto, se in possesso di un titolo noto, dall'abbreviazione del titolo stesso e terminare con una virgola (p.es. "Gentile dott. Mario Rossi,"),. Qualora la comunicazione sia inviata al proponente (Mivoq SRL), essa deve contenere la dicitura "Alla cortese attenzione del dott. Giulio Paci".

Corpo: Il corpo del messaggio deve essere il più chiaro e focalizzato possibile. È desiderabile articolare il contenuto del corpo in punti affinché sia possibile chiarire e rendere massimamente comprensibile il contenuto della comunicazione. È sconsigliato trattare temi non desumibili dall'oggetto della comunicazione.



Formula di chiusura: La formula di chiusura è separata dal corpo con doppio ritorno a capo. Essa prevede il saluto più adatto alla conversazione seguito da una virgola e, dopo un ritorno a capo, il nome del team.

4.1.1.2 Riunioni

4.1.1.2.1 Riunioni interne

Le riunioni interne al team sono fissate ufficialmente ed esclusivamente dal Responsabile di Progetto il quale provvede, una volta stabilito l'ordine del giorno e raccolte le disponibilità dei membri del gruppo, a notificare tutti i membri del *team di sviluppo_G* mediante lo strumento di comunicazione interna. Qualunque membro del team può richiedere al Responsabile di Progetto la convocazione di una riunione interna, a patto di proporre almeno un punto da aggiungere all'ordine del giorno. La convocazione ufficiale verrà effettuata dal Responsabile di Progetto con almeno un giorno di anticipo tramite la piattaforma in uso per le comunicazioni interne. La partecipazione dei membri alle riunioni interne è obbligatoria, in special modo per quelle aventi per tema i rapporti interpersonali tra i membri del *team di sviluppo_G*, salvo problematiche o impegni improrogabili.

4.1.1.2.2 Riunioni esterne

Le riunioni esterne, tenute con il *committente_G* o l'azienda Proponente, saranno concordate dal Responsabile di Progetto che avviserà, con almeno un giorno di preavviso, i membri del team mediante la piattaforma in uso per le comunicazioni interne. Ogni membro del team può richiedere un incontro, specificandone il motivo, al Responsabile di Progetto che deciderà se procedere a contattare la parte esterna. È necessario definire un ordine del giorno affinché la riunione esterna sia il più possibile produttiva.

4.1.1.3 Ticketing

Il sistema di *ticketing_G* disponibile attraverso la piattaforma *Trello_G* permette di assegnare le mansioni ai vari membri del team. Ciò permette ad ogni membro del *team di sviluppo_G* e al Responsabile di Progetto di controllare lo stato di avanzamento di ogni attività necessaria al completamento del progetto. Il Responsabile è l'unico membro del *team di sviluppo_G* che ha il diritto di:

- aprire una card;
- assegnare una card;
- modificare una card;

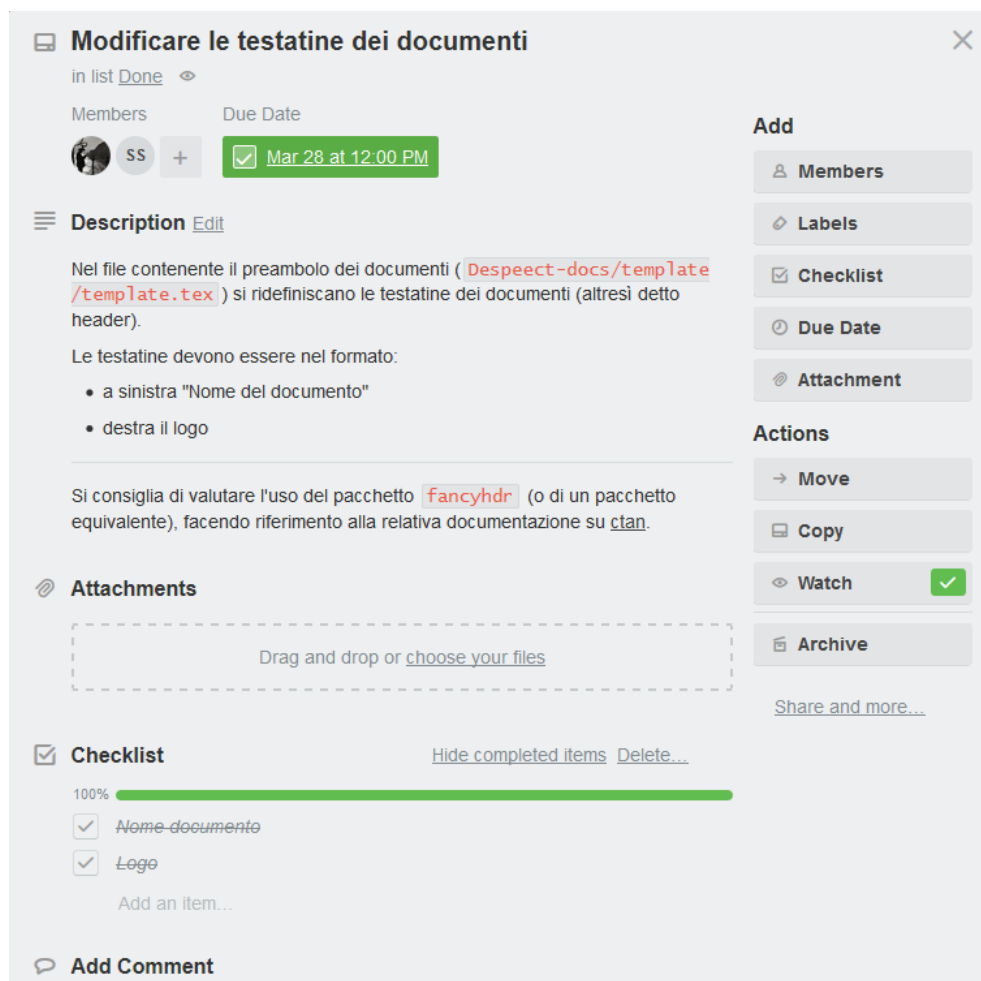


Figura 2: Esempio di card di Trello

- archiviare una card;
- rimuovere una card;
- risolvere una card.

4.1.1.4 Repository

Per permettere la collaborazione e la condivisione dei file necessari allo sviluppo delle varie attività di progetto si è scelto di adottare un *repository*_G. L'host che si è scelto di utilizzare, con licenza "student", permette la creazione di un *repository*_G privato, accessibile ai soli membri del team. L'Amministratore si occupa di inserire i membri del team tra i collaboratori al progetto.



4.1.2 Procedure

4.1.2.1 Apertura di una card

L'apertura di una card consegue da obiettivi, problematiche e necessità emerse durante tutto il progetto. Ogni card dev'essere aperte attraverso l'applicativo *Trello_G* Deve essere definito:

- titolo;
- descrizione esplicativa;
- membri interessati allo svolgimento dei compiti;
- scadenza.

Inoltre è desiderabile che ogni mansione definita tramite sistema di *ticketing_G* sia:

- chiara e non ambigua;
- circoscritta;
- conforme alle attività del progetto.

4.1.2.2 Assegnazione di una card

In seguito all'apertura di una card essa deve essere assegnata. L'assegnazione di una card definisce:

- assegnatari, ovvero i membri del team ch  dovranno completare il compito assegnato.

Inoltre   desiderabile che :

- il compito assegnato sia in linea con il ruolo dell'assegnatario;
- l'assegnatario non sia modificato se non per circostanza eccezionali.

4.1.2.3 Modifica di una card

La modifica di una card   un consegue da modifiche e condizioni eccezionali occorse durante il progetto. Esistono due possibili scenari che possono causare la modifica di una card:

Ritardo: il compito assegnato non   concluso alla data di scadenza stabilit .

Condizioni eccezionali: assenze prolungate che impediscono all'assegnatario di concludere il compito assegnato.



In entrambi i casi il Responsabile di Progetto deve provvedere, non appena possibile, a modificare la card affinché tale modifica impatti nella misura minore possibile l'andamento generale del progetto.

4.1.2.4 Rimozione di un card

La rimozione di una card dipende strettamente da eventi eccezionali occorsi durante tutto l'arco del progetto. È compito del Responsabile di Progetto notificare l'assegnatario del compito della rimozione del card assegnatoli con la massima rapidità possibile.

4.1.2.5 Risoluzione di un card

La risoluzione di una card dipende strettamente dalle attività di verifica e *validazione_G* normalmente presenti nel progetto. Il Responsabile di Progetto, dopo aver ottenuto conferma da parte del Verificatore dell'effettivo completamento della mansione assegnata, provvede a risolvere la card.

4.1.3 Norme

4.1.3.1 Repository

4.1.3.1.1 Nome di File

Il nome di ogni file deve rispettare la seguente formattazione:

NomeDelFile.ext

L'iniziale di ogni parola è maiuscola, non ci sono né spazi né caratteri speciali o accentati.

4.1.3.1.2 Struttura del Repository

È stata definita una *repository_G* per la documentazione di progetto. Tale *repository_G* ha la seguente struttura:

- DocumentiInterni;
 - NormeDiProgetto;
 - StudioDiFattibilità;
 - Verbali.
- DocumentiEsterni;
 - AnalisiDeiRequisiti;



- Glossario;
- PianoDiProgetto;
- PianoDiQualifica.

Tutti i documenti contenuti all'interno della cartella DocumentiInterni sono documenti per uso interno. Tutti i documenti contenuti all'interno della cartella DocumentiEsterni sono documenti per uso esterno.

4.1.3.1.3 Commit

Ogni volta che si terminano le modifiche su un singolo file si deve eseguire una *commit_G* con il comando:

```
git commit -m "messaggio" nomefile
```

Sebbene sia possibile eseguire il *commit_G* di file multipli, si raccomanda di limitarne l'uso. Il messaggio del *commit_G* deve:

- descrivere le modifiche fatte;
- essere conciso (massimo 50 caratteri);
- essere chiaro;
- essere grammaticalmente corretto.

Affinchè sia possibile garantire la massima tracciabilità, è consigliato effettuare *commit_G* dopo aver eseguito modifiche circoscritte e focalizzate.

4.1.4 Strumenti

4.1.4.1 Sistema Operativo

Per l'attività di codifica, i membri del team sono liberi di scegliere una qualsiasi *distribuzione_G* di *GNU/Linux_G* (con *kernel_G* Linux $\geq 4.4.0$). Per le restanti attività, non sono posti vincoli stretti relativamente al sistema operativo scelto.

4.1.4.2 Git

Come sistema di controllo di versione e strumento di *versionamento_G* è stato scelto *Git_G* poichè:

- è software libero;
- è decentralizzato e distribuito;
- documentazione di ottima qualità.



4.1.4.3 Github

GitHub_G è stato scelto come host del *repository_G Git_G*, con licenza studente, in quanto permette l'utilizzo di:

- *repository_G* privati;
- accesso a servizi ulteriori utili allo sviluppo del prodotto;
- *Travis CI_G*, un sistema di *integrazione continua_G*, gratuito per gli studenti;
- un sistema di wiki e di issue interno.

Inoltre *GitHub_G* è ben integrato con il sistema di *ticketing_G Trello_G*.

4.1.4.4 Google Drive

Come strumento di condivisione di documenti non soggetti a *versionamento_G* (p.es. la documentazione di *Speect_G*) è stato scelto *Google Drive_G* poichè:

- è gratuito (fino a 15 GB di spazio);
- è associato implicitamente all'account di posta elettronica creato dal gruppo di lavoro.

4.1.4.5 Trello

Come strumento di gestione di compiti e responsabilità da completare durante il corso del progetto è stato scelto lo strumento *Trello_G* poichè:

- è gratuito;
- ha un'interfaccia pulita e intuitiva;
- è ben integrato con gli altri strumenti in uso per il progetto (p.es. Google Drive, GitHub);
- permette di definire in maniera intuitiva compiti responsabilità e scadenza;
- permette la creazione automatizzata di *diagrammi di Gantt_G*;
- esistono diverse alternative open-source (p.es. *wekan_G*) che permettono di importare le card esportate da *Trello_G*.



4.1.4.6 Telegram

Telegram_G è stato scelto come strumento di messaggistica interna in quanto:

- è sufficientemente leggero da poter essere eseguito anche con dispositivi obsoleti;
- non ci sono limitazioni nella ricerca dei messaggi (a differenza della versione gratuita di Slack);
- sono disponibili diversi *bot_G* (p.es. *IFTTT_G*, *groupagreebot_G*);
- permette di marcare i messaggi con un *hashtag_G*.

4.1.4.7 IFTTT

IFTTT_G è stato scelto per automatizzare alcuni compiti (tra cui l'inoltro delle e-mail dalla casella per le comunicazioni esterne allo strumento di messaggistica interna) in quanto:

- supporta gli strumenti già in uso (p.es. *Telegram_G*, *Trello_G*);
- è di semplice utilizzo.

4.1.4.8 GroupAgree Bot

GroupAgree_G è stato scelto come sistema per le votazioni, per i sondaggi e per la pianificazione delle date delle riunioni, in quanto:

- è integrato con lo strumento di messaggistica interna;
- il voto può essere palese o anonimo;
- permette più tipologie di sondaggio.