

CLIPS

Communication & Localization with Indoor Positioning Systems

UNIVERSITÀ DI PADOVA

PIANO DI QUALIFICA



leaf.gruppo@gmail.com

Versione	1.00
Data Redazione	2016-01-20
Redazione	Andrighetto Cristian Castello Davide
Verifica	Federico Tavella
Approvazione	Federico Tavella
Uso	Esterno
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo Miriade S.p.A.

Diario delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
1.04	2016-02-25	Cristian Andrighetto		Modifica paragrafo 2.1.2.2 - Aggiunta sottoparagrafi da 2.2.1.2.1 a 2.2.1.2.9
1.03	2016-02-21	Cristian Andrighetto	Amministratore	Eliminata sezione 3.3
1.02	2016-02-21	Cristian Andrighetto	Amministratore	Eliminata sezione 2.3
1.01	2016-02-21	Cristian Andrighetto	Amministratore	Modifica sezione 2.2 aggiunto riferimento al Piano di Progetto
1.00	2016-01-20	Federico Tavella	Responsabile di Progetto	Approvazione del documento
0.16	2016-01-18	Federico Tavella	Verificatore	Verifica del resoconto
0.15	2016-01-18	Cristian Andrighetto	Amministratore	Inserimento risultati delle verifiche ai documenti nel resoconto
0.14	2016-01-17	Davide Castello	Amministratore	Stesura resoconto attività di verifica Fase A
0.13	2015-12-27	Federico Tavella	Verificatore	Verifica del documento
0.12	2015-12-26	Davide Castello	Amministratore	Stesura appendice Test
0.11	2015-12-24	Cristian Andrighetto	Amministratore	Stesura appendice CMM

Versione	Data	Autore	Ruolo	Descrizione
0.10	2015-12-23	Davide Castello	Amministratore	Correzione errori sezione strategia in dettaglio
0.09	2015-12-22	Cristian Andrighetto	Amministratore	Correzione errori sezione visione generale della strategia
0.08	2015-12-21	Davide Castello	Amministratore	Riorganizzazione della visione generale della strategia
0.07	2015-12-20	Davide Castello	Amministratore	Fine stesura misure e metriche
0.06	2015-12-17	Davide Castello	Amministratore	Stesura metriche per i documenti
0.05	2015-12-15	Cristian Andrighetto	Amministratore	Stesura sezione riguardante l'organizzazione
0.04	2015-12-13	Davide Castello	Amministratore	Stesura tecniche di controllo della qualità
0.03	2015-12-11	Cristian Andrighetto	Amministratore	Stesura sezione riguardante le risorse
0.02	2015-12-8	Davide Castello	Amministratore	Stesura visione generale della strategia di gestione della qualità
0.01	2015-12-7	Cristian Andrighetto	Amministratore	Stesura struttura documento

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti utili	1
1.4.1	Riferimenti normativi	1
1.4.2	Riferimenti informativi	1
2	Visione generale della strategia di gestione della qualità	3
2.1	Obiettivi di qualità	3
2.1.1	Qualità di processo	3
2.1.1.1	Miglioramento costante - QQPC1	4
2.1.1.2	Rispetto della pianificazione - QQPC2	4
2.1.1.3	Rispetto del budget - QQPC3	5
2.1.2	Qualità di prodotto	5
2.1.2.1	Qualità dei documenti	6
2.1.2.1.1	Leggibilità e comprensibilità - QQ-PRD1	6
2.1.2.1.2	Correttezza ortografica - QQPRD2	6
2.1.2.1.3	Correttezza concettuale - QQPRD3	7
2.1.2.2	Qualità del software	7
2.1.2.2.1	Funzionalità obbligatorie - OBPRS1	8
2.1.2.2.2	Funzionalità desiderabili - OBPRS2	8
2.1.2.2.3	Utilizzo delle funzionalità in modo semplice - OBPRS3	9
2.1.2.2.4	Utilizzo delle funzionalità in modo efficace - OBPRS4	9
2.1.2.2.5	Manutenibilità e Comprensibilità del codice - OBPRS5	10
2.1.2.2.6	Copertura dei test richiesti - OBPRS6	11
2.1.2.2.7	Robustezza - OBPRS7	12
2.1.2.2.8	Funzionamento senza interruzioni - OBPRS8	12
2.1.2.2.9	Installabilità - OBPRS9	13
2.2	Scadenze temporali	13
3	La strategia di gestione della qualità nel dettaglio	14
3.1	Risorse	14
3.1.1	Risorse necessarie	14

3.1.1.1	Risorse umane	14
3.1.1.2	Risorse hardware	14
3.1.1.3	Risorse software	14
3.1.2	Risorse disponibili	15
3.1.2.1	Risorse umane	15
3.1.2.2	Risorse hardware	15
3.1.2.3	Risorse software	15
3.2	Tecniche di controllo della qualità	15
3.2.1	Tecniche di controllo della qualità di processo	15
3.2.2	Tecniche di controllo della qualità di prodotto	16
3.2.2.1	Verifica	16
3.2.2.1.1	Analisi statica	16
3.2.2.1.2	Analisi dinamica	16
3.2.2.2	Validazione	17
3.3	Misure e metriche	17
3.3.1	Misure	17
3.3.2	Metriche per i processi	17
3.3.2.1	Capability Maturity Model - MPC1	18
3.3.2.2	Schedule Variance - MPC2	18
3.3.2.3	Budget Variance - MPC3	18
3.3.3	Metriche per i prodotti	19
3.3.3.1	Metriche per i documenti	19
3.3.3.1.1	Indice di leggibilità - MPRD1	19
3.3.3.1.2	Errori ortografici rinvenuti e non corretti - MPRD2	20
3.3.3.1.3	Errori concettuali rinvenuti e non corretti - MPRD3	20
3.3.3.2	Metriche per il software	21
3.3.3.2.1	Copertura Requisiti Obbligatori - MPRS1	22
3.3.3.2.2	Copertura Requisiti Desiderabili - MPRS2	22
3.3.3.2.3	Numero di Passaggi per Operazione - MPRS3	22
3.3.3.2.4	Function Understandability - MPRS4	23
3.3.3.2.5	Task Completion - MPRS5	23
3.3.3.2.6	Numero di statement per metodo - MPRS6	23
3.3.3.2.7	Numero di campi dati per classe - MPRS7	24
3.3.3.2.8	Grado di accoppiamento - MPRS8	24
3.3.3.2.9	Cyclomatic Number - MPRS9	24
3.3.3.2.10	Conditional Statement - MPRS10	25

3.3.3.2.11	Adequacy of variable names - MPRS11	25
3.3.3.2.12	Average Module Size - MPRS12 . . .	25
3.3.3.2.13	Test Passati Richiesti - MPRS13 . .	26
3.3.3.2.14	Failure Avoidance - MPRS14	26
3.3.3.2.15	Breakdown Avoidance - MPRS15 . .	26
3.3.3.2.16	Numero di installazioni completate - MPRS15	27
A	Capability Maturity Model	28
A.1	Struttura	28
A.2	Livelli	29
B	Standard ISO/IEC 9126	31
B.1	Modello della qualità del software	31
B.1.1	Modello della qualità interna ed esterna	31
B.1.2	Modello della qualità in uso	32
B.2	Metriche per la qualità interna	33
B.3	Metriche per la qualità esterna	33
B.4	Metriche per la qualità in uso	33
C	PDCA	34
D	Test	35
D.1	Test di accettazione	35
D.2	Test di sistema	35
E	Resoconto delle attività di verifica - Fase A	36
E.1	Resoconto delle attività di verifica sui prodotti	36
E.1.1	Documenti	36
E.1.1.1	Verifiche manuali	36
E.1.1.2	Verifiche automatiche	37
E.2	Resoconto delle attività di verifica sui processi	38
E.2.1	Processo di documentazione	38
E.2.1.1	Livello CMM	38
E.2.2	Processo di verifica	39
E.2.2.1	Livello CMM	39

Elenco delle tabelle

1	Mappa Metriche-Caratteristiche	21
2	Errori trovati tramite verifica manuale dei documenti durante la fase _g A	37
3	Nuovi termini da inserire nel <i>Glossario</i> individuati tramite verifica manuale dei documenti durante la fase _g A	37
4	Errori trovati tramite verifica automatica dei documenti durante la fase _g A	38
5	Esiti del calcolo dell'indice di leggibilità effettuato tramite strumenti automatici durante la fase _g A	38

1 Introduzione

1.1 Scopo del documento

Il presente documento ha l'obiettivo di fissare gli obiettivi di qualità e le strategie che il gruppo *Leaf* ha deciso di adottare per perseguirli. Questo documento darà inoltre una visione di come il gruppo affronterà le varie fasi di verifica per poter conseguire il miglior risultato possibile in termini di qualità.

1.2 Scopo del prodotto

Lo scopo del prodotto_g è implementare un metodo di navigazione indoor_g che sia funzionale alla tecnologia BLE_g. Il prodotto_g comprenderà un prototipo software_g che permetta la navigazione all'interno di un'area predefinita, basandosi sui concetti di IPS_g e smart places_g.

1.3 Glossario

Allo scopo di rendere più semplice e chiara la comprensione dei documenti viene allegato il *Glossario v1.00* nel quale verranno raccolte le spiegazioni di terminologia tecnica o ambigua, abbreviazioni ed acronimi. Per evidenziare un termine presente in tale documento, esso verrà marcato con il pedice _g.

1.4 Riferimenti utili

1.4.1 Riferimenti normativi

- **Norme di Progetto:** *Norme di progetto v1.00*;
- **Standard [ISO/IEC 9126:2001]:**
https://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Capability Maturity Model (CMM_g):**
https://en.wikipedia.org/wiki/Capability_Maturity_Model;
- **Plan-Do-Check-Act (PDCA_g):**
<https://en.wikipedia.org/wiki/PDCA>.

1.4.2 Riferimenti informativi

- **Piano di Progetto:** *Piano di progetto v1.00*;

- indice Gulpease_g: https://it.wikipedia.org/wiki/Indice_Gulpease;
- Slide del corso di Ingegneria del software_g - Qualità del software_g:
<http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/L08.pdf>;
- Slide del corso di Ingegneria del software_g - Qualità del processo_g:
<http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/L09.pdf>.

2 Visione generale della strategia di gestione della qualità

2.1 Obiettivi di qualità

In questa sezione vengono riportati gli obiettivi di qualità che il gruppo *Leaf* si impegna a perseguire durante lo svolgimento dell'intero progetto. Per capire se un certo obiettivo è stato raggiunto oppure no, il gruppo fa uso di standard, modelli e metriche. Ognuno di questi fa uso di scale differenti e fissate a priori: qualunque sia il criterio utilizzato per misurare e dunque quantificare la vicinanza a un certo obiettivo abbiamo fissato dei valori minimi che intendiamo raggiungere nell'arco dell'intero progetto. Oltre a ciò abbiamo anche fissato dei valori che riteniamo ottimali e che devono essere sperabilmente (ma non obbligatoriamente) raggiunti.

I seguenti obiettivi sono inoltre identificati da un codice identificativo, al fine di rendere più semplice il tracciamento tra l'obiettivo di qualità da soddisfare e la metrica che permettere di verificarne il soddisfacimento.

La procedura di denominazione degli obiettivi è spiegata in dettaglio nel documento *Norme di progetto v1.00*.

2.1.1 Qualità di processo

Assicurare la qualità dei processi è indispensabile durante lo svolgimento del progetto per le seguenti ragioni:

- aiuta ad ottimizzare l'uso delle risorse;
- fa in modo che i costi siano maggiormente contenuti;
- migliora la stima dei rischi e degli impegni.

Un altro fattore da tenere in considerazione è che molto spesso prodotti scadenti derivano da processi scadenti.

Le caratteristiche ottimali che desideriamo che i processi posseggano vengono riportate in seguito:

- un processo dovrebbe essere in grado di migliorare continuamente le proprie performance
 - le performance di un processo dovrebbero essere costantemente misurabili;
 - un processo, durante il proprio svolgimento, dovrebbe cercare di perseguire sempre degli obiettivi quantitativi di miglioramento fissati a priori.

- le attività di un processo dovrebbero proseguire nei tempi indicati nel documento *Piano di progetto v1.00*;
- i costi effettivi di ogni processo dovrebbero essere in linea con quanto dichiarato nel documento *Piano di progetto v1.00*.

Nelle prossime sezioni si enunciano gli obiettivi che il gruppo *Leaf* intende raggiungere. Per ognuno di essi vengono specificati i criteri con i quali si effettuano le misurazioni sulla qualità (per capire quanto si è vicini all'obiettivo).

Per ogni criterio scelto vengono inoltre dichiarati i valori minimi che si intendono raggiungere, oltre a quelli ottimali.

2.1.1.1 Miglioramento costante - OQPC1 Per misurare quanto si è vicini all'obiettivo di avere processi in grado di misurare le proprie performance e che sono quindi in grado di porsi obiettivi quantitativi di miglioramento si è deciso di adottare il modello CMM.

In particolare si vuole raggiungere almeno il livello 2 previsto da tale scala. Il livello ottimale che sperabilmente dovremmo raggiungere è il 4.

Riassumendo:

Modello utilizzato: CMM;

Soglia di accettabilità: livello 2 previsto da CMM;

Soglia di ottimalità: livello 4 previsto da CMM;

Per una migliore e più dettagliata descrizione del modello CMM qui adottato si faccia riferimento all'appendice [A](#) "Capability Maturity Model".

Per approfondire la scelta dei range di accettazione e ottimalità si consulti invece la metrica MPC1 "Capability Maturity Model" alla sezione [3.3.2.1](#).

2.1.1.2 Rispetto della pianificazione - OQPC2 Per capire se le attività di un processo sono oppure no in ritardo rispetto a quanto pianificato all'interno del *Piano di progetto* viene utilizzata la seguente metrica: Schedule Variance.

Si desidera che il ritardo accumulato sia minore del 5% rispetto al totale pianificato. Sarebbe invece ottimale essere esattamente in linea con quanto prevede il *Piano di progetto*, o essere addirittura in anticipo.

Riassumendo:

Metrica utilizzata: Schedule Variance;

Soglia di accettabilità: in ritardo al massimo del 5% rispetto a quanto pianificato;

Soglia di ottimalità: essere in linea o in anticipo con quanto pianificato (ritardo minore o uguale a 0%).

Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica MPC2 "Schedule Variance" alla sezione [3.3.2.2](#).

2.1.1.3 Rispetto del budget - OQPC3 Per capire se i costi di un processo rientrano oppure no nel budget previsto dal *Piano di progetto* viene utilizzata la seguente metrica: Budget Variance.

L'obiettivo minimo è quello di avere dei costi che non superano il budget a disposizione per più del 10%. Sarebbe invece ottimale che i costi fossero esattamente in linea con il preventivo o che addirittura si avesse speso meno. Riassumendo:

Metrica utilizzata: Budget Variance;

Soglia di accettabilità: costi al massimo maggiori del 10% rispetto al preventivo;

Soglia di ottimalità: costi in linea o minori di quanto previsto dal *Piano di progetto* (minore o uguale a 0%).

Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica MPC3 "Budget Variance" alla sezione [3.3.2.3](#).

2.1.2 Qualità di prodotto

Il gruppo si prefigge di mantenere la stessa qualità sia nei processi che nei prodotti: per garantire la migliore qualità del prodotto_g anche il processo_g da cui proviene deve avere una buona qualità. Il gruppo per mantenere la qualità del prodotto_g cercherà di seguire al meglio lo standard di qualità [ISO/IEC 9126:2001].

I prodotti che vengono realizzati durante l'intero progetto sono sostanzialmente di due tipi: documenti e software_g. Nelle prossime sezioni, si enunciano gli obiettivi che il gruppo *Leaf* intende raggiungere suddivisi per tipologia di prodotto_g.

Per ogni obiettivo, poi, vengono specificati i criteri con i quali si effettuano le misurazioni sulla qualità (per capire quanto si è vicini all'obiettivo).

Per ogni criterio scelto vengono inoltre dichiarati i valori minimi che si intendono raggiungere, oltre a quelli ottimali.

2.1.2.1 Qualità dei documenti Gli obiettivi di qualità riguardanti i documenti ai quali il gruppo *Leaf* desidera arrivare nell'arco dell'intero progetto sono i seguenti:

- i documenti devono essere comprensibili da individui dotati di una licenza superiore;
- i documenti devono essere corretti a livello ortografico;
- i documenti non devono contenere concetti errati.

Descriviamo ora quali sono le metriche o i criteri che si intendono utilizzare per quantificare la vicinanza a ognuno degli obiettivi sopra descritti.

Individuiamo inoltre le soglie di accettabilità e ottimalità, per fissare quantitativamente i punti ai quali desideriamo arrivare.

2.1.2.1.1 Leggibilità e comprensibilità - OQPRD1 Per cercare di capire quanto i documenti siano effettivamente leggibili e comprensibili da persone dotate di una licenza superiore viene utilizzato l'indice Gulpease_g. Si desidera che i documenti posseggano costantemente un indice maggiore a 40 (soglia di accettabilità). Si dovrebbe tuttavia cercare di raggiungere un valore più alto, considerato ottimale, ovvero 60.

Riassumendo:

Metrica utilizzata: indice Gulpease_g;

Soglia di accettabilità: valori almeno maggiori di 40;

Soglia di ottimalità: valori almeno maggiori di 60.

Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica MPRD1 "Indice di leggibilità" alla sezione [3.3.3.1.1](#).

2.1.2.1.2 Correttezza ortografica - OQPRD2 Per capire quanto i documenti siano effettivamente corretti a livello ortografico viene utilizzata la seguente metrica: percentuale di errori ortografici rinvenuti e non corretti. Si desidera che tutti gli errori ortografici che sono stati trovati siano corretti. In questo caso, dunque, l'obiettivo minimo coincide con l'obiettivo ottimale. Riassumendo:

Metrica utilizzata: percentuale di errori ortografici rinvenuti e non corretti;

Soglia di accettabilità: tutti gli errori trovati sono stati corretti;

Soglia di ottimalità: tutti gli errori trovati sono stati corretti.

Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica MPRD2 "Errori ortografici rinvenuti e non corretti" alla sezione [3.3.3.1.2](#).

2.1.2.1.3 Correttezza concettuale - OQPRD3 Per capire quanto i documenti siano effettivamente corretti a livello concettuale viene utilizzata la seguente metrica: percentuale di errori concettuali rinvenuti e non corretti. Si desidera che al massimo il 5% degli errori concettuali rinvenuti non siano corretti. L'obiettivo ottimale sarebbe quello di avere documenti senza alcun errore concettuale.

Riassumendo:

Metrica utilizzata: percentuale di errori concettuali rinvenuti e non corretti;

Soglia di accettabilità: almeno il 95% degli errori trovati è stato corretto;

Soglia di ottimalità: tutti gli errori trovati sono stati corretti.

Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica MPRD3 "Errori concettuali rinvenuti e non corretti" alla sezione [3.3.3.1.3](#).

2.1.2.2 Qualità del software Gli obiettivi di qualità del software, ai quali il gruppo *Leaf* desidera arrivare nell'arco del progetto sono alcuni di quelli che sono enunciati all'interno dello standard [ISO/IEC 9126:2001]. Vengono riassunti in seguito:

- il prodotto_g possiede le funzionalità descritte all'interno dei requisiti obbligatori;
- il prodotto_g possiede le funzionalità descritte all'interno dei requisiti desiderabili;
- il prodotto_g permette agli utenti di utilizzare le funzionalità in maniera semplice;
- il prodotto_g permette agli utenti di utilizzare le funzionalità in maniera efficace;
- il codice risulta manutenibile e facilmente comprensibile;

- il prodotto_g è testato in ogni sua parte e in ogni situazione nella quale si può trovare;
- il prodotto_g è robusto e non interrompe l'esecuzione in seguito a situazioni anomale;
- il prodotto_g garantisce un funzionamento senza interruzioni;
- il prodotto_g è facilmente installabile.

2.1.2.2.1 Funzionalità obbligatorie - OBPRS1 Il prodotto deve ricoprire tutte le funzionalità descritte nei requisiti obbligatori. Per monitorare lo stato di completamento delle funzionalità richieste il gruppo ha pensato di rapportare i requisiti completati con quelli ancora da completare.

Metrica utilizzata: Copertura Requisiti Obbligatori

Soglia di accettabilità: 100% requisiti soddisfatti

Soglia di ottimalità: 100% requisiti soddisfatti

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS1 "Copertura Requisiti Obbligatori" alla sezione [3.3.3.2.1](#).

2.1.2.2.2 Funzionalità desiderabili - OBPR2 Il prodotto deve ricoprire tutte le funzionalità descritte nei requisiti desiderabili. Per monitorare lo stato di completamento delle funzionalità richieste il gruppo ha pensato di rapportare i requisiti completati con quelli ancora da completare.

Metrica utilizzata: Copertura Requisiti Desiderabili

Soglia di accettabilità: 100% requisiti soddisfatti

Soglia di ottimalità: 100% requisiti soddisfatti

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS2 "Copertura Requisiti Desiderabili" alla sezione [3.3.3.2.2](#).

2.1.2.2.3 Utilizzo delle funzionalità in modo semplice - OBPR3

Il prodotto deve offrire delle funzionalità semplici all'utilizzo, quindi il numero di passaggi da compiere per completare un'operazione deve essere minimo. Il gruppo ritiene che dei valori abbastanza bassi possano soddisfare l'obiettivo. Per valutare la semplicità delle funzionalità del prodotto si considera anche la percentuale di funzionalità capite da parte dell'utente.

Metrica utilizzata: Numero di Passaggi per Operazione

Soglia di accettabilità: $5 < X < 8$

Soglia di ottimalità: $X < 5$

X = numero di passaggi per effettuare un'operazione

Metrica utilizzata: Function Understandability

Soglia di accettabilità: $70\% < X < 80\%$

Soglia di ottimalità: $X > 80\%$

X = percentuale di funzioni capite correttamente su funzioni totali

Per una descrizione dettagliata delle metriche qui utilizzate e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alle sezioni

- [3.3.3.2.3](#) - MPRS3 - Numero di Passaggi per Operazione
- [3.3.3.2.4](#) - MPRS4 - Function Understandability

2.1.2.2.4 Utilizzo delle funzionalità in modo efficace - OBPR4

Il prodotto deve permettere di portare a termine correttamente buona parte delle azioni svolte dall'utente. Per questo si cerca di valutare in media la percentuale di task completati da un utente.

Metrica utilizzata: Task Completion

Soglia di accettabilità: $70\% < X < 80\%$

Soglia di ottimalità: $X > 80\%$

X = percentuale media di task completati su task assegnati ad un utente

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS5 "Task Completion".

2.1.2.2.5 Manutenibilità e Comprensibilità del codice - OBPR5

Il prodotto deve avere codice manutenibile e non deve presentare incomprensioni al suo interno. Per questo si tiene conto della sua complessità e della sua lunghezza. Codice poco manutenibile può portare all'abbandono dello sviluppo del prodotto.

Metrica utilizzata: Numero di statement per metodo

Soglia di accettabilità: $30 < X < 40$

Soglia di ottimalità: $X < 30$

X = numero di statement per metodo

Metrica utilizzata: Numero di campi dati per classe

Soglia di accettabilità: $10 < X < 15$

Soglia di ottimalità: $X < 10$

X = numero di campi dati per classe

Metrica utilizzata: Grado di accoppiamento

Soglia di accettabilità: $3 < X < 7$

Soglia di ottimalità: $X < 3$

X = numero di dipendenze tra classi in un package_g

Metrica utilizzata: Cyclomatic Number

Soglia di accettabilità: $4 < X < 10$

Soglia di ottimalità: $X < 4$

X = numero di complessità ciclomantica

Metrica utilizzata: Conditional Statement

Soglia di accettabilità: $20 < X < 50$

Soglia di ottimalità: $X < 20$

X = numero di statement condizionali in un modulo

N = numero di moduli

Metrica utilizzata: Adequacy of variable names

Soglia di accettabilità: $80\% < X < 90\%$

Soglia di ottimalità: $X > 90\%$

X = percentuale dei nomi delle variabili che corrispondono alla *Definizione di prodotto v1.00*

Metrica utilizzata: Adequacy Module Size

Soglia di accettabilità: $300 < X < 400$

Soglia di ottimalità: $200 < X < 300$

X = numero di linee di codice per modulo

Per una descrizione dettagliata delle metriche qui utilizzate e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alle sezioni

- [3.3.3.2.6](#) - MPRS6 - Numero di statement per metodo
- [3.3.3.2.7](#) - MPRS7 - Numero di campi dati per classe
- [3.3.3.2.8](#) - MPRS8 - Grado di accoppiamento
- [3.3.3.2.9](#) - MPRS9 - Cyclomatic Number
- [3.3.3.2.10](#) - MPRS10 - Conditional Statement
- [3.3.3.2.11](#) - MPRS11 - Adequacy of variable names
- [3.3.3.2.12](#) - MPRS12 - Adequacy Module Size

2.1.2.2.6 Copertura dei test richiesti - OBPRS6 Il prodotto deve essere testato in ogni sua parte per garantirne il suo funzionamento. I test presi in considerazione sono quelli che testano le funzionalità previste dai requisiti.

Metrica utilizzata: Test Passati Richiesti

Soglia di accettabilità: $80\% < X < 90\%$

Soglia di ottimalità: $90\% < X < 98\%$

X = percentuale di test passati su test ricavati dai requisiti

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS13 "Test Passati Richiesti" nella sezione [3.3.3.2.13](#).

2.1.2.2.7 Robustezza - OBPRS7 Il prodotto deve essere robusto e non deve interrompere il suo funzionamento in seguito a situazioni anomale presentate. Il prodotto deve essere in grado inoltre di gestire le situazioni di errore.

Metrica utilizzata: Failure Avoidance

Soglia di accettabilità: $80\% < X < 90\%$

Soglia di ottimalità: $X > 90\%$

X = percentuale di situazioni anomale evitate su situazioni anomale prese in considerazione

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS14 "Failure Avoidance" nella sezione [3.3.3.2.14](#).

2.1.2.2.8 Funzionamento senza interruzioni - OBPRS8 Il prodotto deve garantire un funzionamento senza interruzioni. Questo livello è considerato ottimale ma sono accettati anche valori maggiori dell'80%.

Metrica utilizzata: Breakdown Avoidance

Soglia di accettabilità: $80\% < X < 90\%$

Soglia di ottimalità: $X > 90\%$

X = percentuale di interruzioni evitate in base a situazioni anomale presentate

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS15 "Breakdown Avoidance" nella sezione [3.3.3.2.15](#).

2.1.2.2.9 Installabilità - OBPRS9 Il prodotto, per essere utilizzato deve essere facilmente installabile quindi è prevista una metrica che valuta l'installabilità del prodotto.

Numero di installazioni completate

Soglia di accettabilità: $80\% < X < 90\%$

Soglia di ottimalità: $X > 90\%$

X = percentuale di installazioni completate su installazioni provate

Per una descrizione dettagliata della metrica qui utilizzata e per una maggiore comprensione degli indici di ottimalità e accettabilità presentati si faccia riferimento alla metrica MPRS16 "Numero di installazioni completate" nella sezione [3.3.3.2.16](#).

2.2 Scadenze temporali

Le scadenze che il gruppo *Leaf* ha deciso di rispettare sono riportate nel *Piano di progetto v1.00*.

3 La strategia di gestione della qualità nel dettaglio

3.1 Risorse

Per garantire un buon funzionamento del processo_g di verifica verranno impiegate diverse risorse.

Le risorse si suddivideranno in:

- risorse umane;
- risorse hardware;
- risorse software_g.

3.1.1 Risorse necessarie

3.1.1.1 Risorse umane Le risorse umane di cui il processo_g di verifica avrà bisogno sono il *Responsabile di progetto* e i *Verificatori*. Informazioni più dettagliate sui ruoli sono riportate nelle *Norme di progetto v1.00*. Le responsabilità che ricadono su queste due figure sono riportate alla sezione 3.3 "Organizzazione" del presente documento.

3.1.1.2 Risorse hardware Per eseguire la verifica il gruppo dovrà avere a disposizione dei computer con un'adeguata potenza di calcolo in grado di sopportare il carico di lavoro.

3.1.1.3 Risorse software Le risorse software_g necessarie alla verifica sono gli strumenti software_g che permettono di eseguire controlli sui documenti e verificare che essi aderiscano alle *Norme di progetto v1.00*.

Gli strumenti software_g dovranno avere le seguenti caratteristiche:

- rilevare (durante la scrittura) eventuali errori ortografici;
- costruire e visualizzare in tempo reale il documento scritto in L^AT_EX_g (in modo che sia facile accorgersi di errori nell'utilizzo dei comandi).

Inoltre è necessario disporre di una piattaforma che raccolga i vari errori incontrati e li segnali ai componenti del gruppo che dovranno occuparsene.

3.1.2 Risorse disponibili

3.1.2.1 Risorse umane Il gruppo ha a disposizione tutti i membri per eseguire operazioni di verifica. A turno ognuno dei componenti ricoprirà il ruolo di *Responsabile di progetto* o di *Verificatore* come definito nel *Piano di progetto v1.00*.

3.1.2.2 Risorse hardware Le risorse hardware disponibili sono i vari computer dei componenti del gruppo incaricati di svolgere il ruolo di *Responsabile di progetto* o *Verificatore*. Eventualmente sono disponibili anche i computer del Servizio Calcolo dell'Università di Padova.

3.1.2.3 Risorse software Le risorse software_g disponibili sono editor L^AT_EX_g con controlli integrati e script per controllare la leggibilità e la complessità dei documenti in riferimento all'indice Gulpease_g. Sarà disponibile anche il sistema di sollevamento delle issue_g offerto dalla piattaforma GitHub_g; per maggiori informazioni sulla procedura di sollevamento e gestione delle issue_g si veda il documento *Norme di progetto v1.00*.

3.2 Tecniche di controllo della qualità

3.2.1 Tecniche di controllo della qualità di processo

Il gruppo intende misurare con continuità le caratteristiche di qualità dei vari processi al fine di apportare miglioramenti in modo sistematico. Per mettere in atto ciò ci si basa su quanto descritto in seguito.

- Ci si basa innanzitutto sul modello CMM_g, quindi sui concetti di "capability" e "maturity" (vedi appendice A "Capability Maturity Model").
- Grazie all'uso del modello CMM_g è possibile calcolare il livello di maturità di un processo_g per confrontare le performance dello stesso in momenti differenti: in questo modo si possono evidenziare eventuali miglioramenti raggiunti (secondo quanto descritto nel presente documento alla sezione 2 "Misure e metriche").
- Per misurare la qualità di un processo_g può essere utile verificare quella del suo prodotto_g: se essa è scarsa, ciò implica che probabilmente anche il processo_g dal quale deriva non è per nulla di qualità.
- Per ottenere un miglioramento continuo dei processi si utilizza il modello del miglioramento continuo PDCA_g.

3.2.2 Tecniche di controllo della qualità di prodotto

3.2.2.1 Verifica Quando si effettuano delle verifiche si usano tanto tecniche di analisi statica quanto di analisi dinamica. L'analisi statica può essere applicata sia alla documentazione che al software_g, mentre l'analisi dinamica solamente al software_g. Il gruppo ha deciso di adottare le tecniche di controllo della qualità descritte in seguito.

3.2.2.1.1 Analisi statica

Inspection Questa tecnica di analisi presuppone l'esperienza da parte del verificatore nell'individuare gli errori e le anomalie più frequenti. A tal scopo, quando il gruppo riterrà di avere l'esperienza necessaria verrà stilata una lista di controllo nella quale saranno elencati gli errori comuni. Questo ci consentirà una verifica più rapida, impegnando meno risorse umane.

Walkthrough Questa tecnica di analisi prevede una lettura critica del codice o del documento prodotto_g. Tale tecnica è molto dispendiosa in termini di risorse, poiché viene applicata all'intero documento senza avere una precisa idea di quale sia il tipo di anomalia_g e di dove ricercarla;

3.2.2.1.2 Analisi dinamica

Test di unità Consiste nel verificare ogni singola unità del prodotto software_g, ovvero sia la più piccola parte di software_g che conviene testare da sola, attraverso l'utilizzo di strumenti come logger_g, stub_g o driver_g. Data la sua natura, la dimensione dell'unità da testare verrà definita al momento del test. Lo scopo dello unit testing_g è di verificare il corretto funzionamento di un'unità per permettere una precoce individuazione dei bug_g. Uno unit testing_g accurato produce vari vantaggi, ad esempio:

- semplifica le modifiche;
- semplifica l'integrazione;
- supporta la documentazione.

Test di integrazione Consiste nella verifica dei componenti del sistema che sono formati dalla combinazione di più unità. Ha lo scopo di evidenziare gli eventuali errori residui non individuati durante la realizzazione dei singoli moduli.

Test di sistema Consiste nell'eseguire nuovamente i test di unità e integrazione per le componenti che hanno subito modifiche o per le nuove funzionalità. Lo scopo è verificare di avere un prodotto_g di alta qualità per ogni nuova funzionalità o modifica importante e di soddisfare tutti i requisiti software_g previsti.

Test di accettazione È il collaudo del prodotto software_g che viene eseguito in presenza del proponente a dimostrazione del fatto che il prodotto software_g soddisfa tutti i requisiti utente. Se tale collaudo viene superato positivamente si può procedere al rilascio ufficiale del prodotto sviluppato.

3.2.2.2 Validazione La validazione avviene nel momento in cui il prodotto_g ha superato i test di verifica ed è pronto al suo rilascio. Il prodotto_g dopo aver superato la validazione ci conferma che è conforme alle aspettative e soddisfa tutti i requisiti, di conseguenza è pronto per essere rilasciato.

3.3 Misure e metriche

3.3.1 Misure

Ogni volta che viene effettuata una misura sui processi o sui prodotti essa va rapportata in una scala. Di seguito vengono riportati i valori della scala:

Negativo valore non accettabile, bisogna effettuare ulteriori verifiche e correggere gli errori presenti.

Accettabile valore accettabile, l'oggetto sottoposto a verifica ha raggiunto una soglia minima.

Ottimale valore accettabile, l'oggetto sottoposto a verifica ha raggiunto le massime aspettative del team_g. L'obiettivo dovrebbe essere quello di avere tutti i valori all'interno di tale range.

3.3.2 Metriche per i processi

Le seguenti metriche sono identificate da un codice identificativo, al fine di rendere più semplice il tracciamento tra l'obiettivo di qualità da soddisfare e la metrica che permettere di verificarne il soddisfacimento.

La procedura di denominazione delle metriche è spiegata in dettaglio nel documento *Norme di progetto v1.00*.

3.3.2.1 Capability Maturity Model - MPC1 Per controllare e verificare la qualità dei processi, il gruppo adotterà le metriche fornite dal modello CMM_g dove per ogni fase_g di lavoro si andrà a fornire un indice che descriverà la qualità della fase_g presa in esame. L'indice sarà relativo ad una scala già definita dal CMM_g.

Effettuando questo tipo di verifiche il team_g avrà subito un riscontro della qualità del processo_g. CMM_g ci consente di individuare la maturità di un processo_g: essa può assumere un valore da 1 (il peggiore) a 5 (il migliore). Mettendo ora in relazione i risultati di tale modello con i range da noi stabiliti otteniamo quanto segue:

- il valore 1 è considerato negativo;
- i valori 2 e 3 sono considerati accettabili;
- i valori 4 e 5 sono considerati ottimali.

3.3.2.2 Schedule Variance - MPC2 La presente metrica indica se le attività di progetto sono in anticipo o in ritardo rispetto a quelle pianificate nel *Piano di progetto*.

Costituisce un indicatore di efficacia dei processi e viene calcolata come la differenza fra la data pianificata di fine di un'attività e la data di fine reale dell'attività stessa.

Se la schedule variance è minore di 0 significa che il progetto sta producendo con minor velocità rispetto a quanto pianificato, viceversa se positivo. Se è pari a 0 significa che si è perfettamente in linea con la pianificazione.

I range di accettazione per questa metrica sono:

- un deficit maggiore del 5% del tempo pianificato per il processo è considerato negativo;
- un deficit minore del 5% del tempo pianificato per il processo è considerato accettabile;
- valori maggiori o uguali a 0 sono considerati ottimali.

3.3.2.3 Budget Variance - MPC3 La presente metrica indica se alla data corrente i costi sono maggiori o minori rispetto a quanto previsto.

Costituisce un indice di efficienza e si calcola confrontando il preventivo con il consuntivo.

I range di accettazione per questa metrica sono:

- un deficit maggiore del 10% delle risorse preventivate per il processo è considerato negativo;
- un deficit minore del 10% delle risorse preventivate per il processo è accettabile;
- un valore maggiore o uguale a 0 è considerato ottimale.

3.3.3 Metriche per i prodotti

Le seguenti metriche sono identificate da un codice identificativo, al fine di rendere più semplice il tracciamento tra l'obiettivo di qualità da soddisfare e la metrica che permettere di verificarne il soddisfacimento.

La procedura di denominazione delle metriche è spiegata in dettaglio nel documento *Norme di progetto v1.00*.

3.3.3.1 Metriche per i documenti La qualità di un documento dipende prima di tutto dai suoi contenuti. La loro qualità, tuttavia, è difficilmente quantificabile allo stato attuale del progetto a causa dell'esperienza pressoché nulla del team_g in quest'ambito. Si è deciso dunque di limitarsi a valutare parametri maggiormente oggettivi e soprattutto misurabili automaticamente attraverso strumenti software_g.

3.3.3.1.1 Indice di leggibilità - MPRD1 Una metrica che si è deciso di utilizzare per poter stimare la qualità di un documento è l'indice di leggibilità. In particolare, è stato considerato l'indice Gulpease_g, studiato appositamente per la lingua italiana.

Questo particolare indice si basa sulla lunghezza della parola e sulla lunghezza della frase rispetto al numero di lettere. La formula per il suo calcolo è la seguente:

$$Indice\ Gulpease = 89 + \frac{300 * numeroFrase - 10 * numeroLettere}{numeroParole} \quad (1)$$

Il risultato di tale equazione tipicamente è compreso tra 0 e 100, dove valori alti indicano leggibilità elevata e viceversa.

In generale, risulta che testi con un indice:

- inferiore a 80 risultano difficili da leggere per chi ha la licenza elementare;
- inferiore a 60 risultano difficili da leggere per chi ha la licenza media;
- inferiore a 40 risultano difficili da leggere per chi ha la licenza superiore.

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta. Si noti che viene tenuto in considerazione il fatto che la documentazione è destinata a persone sufficientemente preparate, competenti ed istruite.

- Valori minori di 40 sono considerati negativi.
- Valori compresi tra 40 e 60 sono considerati accettabili.
- Valori maggiori di 60 sono considerati ottimali.

3.3.3.1.2 Errori ortografici rinvenuti e non corretti - MPRD2

Tale metrica è necessaria per capire quanto un documento sia corretto dal punto di vista ortografico. Infatti, supponendo che gli strumenti automatici siano in grado di trovare tutti (o perlomeno la maggior parte) degli errori ortografici all'interno di un testo, la correttezza ortografica non può che basarsi sul numero di errori rinvenuti ma non successivamente corretti. Notare che per errori corretti si intende un errore revisionato manualmente da parte di un *Verificatore*. Le correzioni automatiche, infatti, non sono molto attendibili.

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale di errori non corretti maggiore allo 0% è ritenuta negativa;
- una percentuale di errori non corretti pari allo 0% è ritenuta accettabile;
- una percentuale di errori non corretti pari allo 0% è ritenuta ottimale.

Notare che non è accettabile che vi siano errori rinvenuti e non corretti da qualche membro del gruppo.

3.3.3.1.3 Errori concettuali rinvenuti e non corretti - MPRD3

Tale metrica è necessaria per capire quanto un documento sia corretto dal punto di vista concettuale. Infatti, supponendo che in seguito alle revisioni siano stati trovati tutti (o perlomeno la maggior parte) i maggiori errori di questo tipo, la correttezza concettuale non può che basarsi sul numero di errori rinvenuti e fatti notare ma non successivamente corretti. Notare che per errori corretti si intende un errore fatto notare dal committente o da qualche *Verificatore* (con relativa approvazione del *Responsabile di progetto*) e successivamente corretto (sulla base di discussioni interne o con il committente).

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale di errori non corretti maggiore al 5% è ritenuta negativa;
- una percentuale di errori non corretti minore del 5% è ritenuta accettabile;
- una percentuale di errori non corretti pari allo 0% è ritenuta ottimale;

3.3.3.2 Metriche per il software Il gruppo *Leaf* per garantire la qualità del software ha deciso di adottare delle metriche. Esse hanno il compito di monitorare la qualità interna, qualità esterna e la qualità in uso. In base alle risorse a disposizione e agli obiettivi di qualità del software preposti il gruppo ha deciso di adottare alcune metriche presenti all'interno dello standard [ISO/IEC 9126:2001].

Ogni metrica scelta viene associata ad una caratteristica di qualità presente all'interno dello standard:

Metriche scelte	Caratteristiche di Qualità
MPRS1 - Copertura Requisiti Obbligatorii	Funzionabilità
MPRS2 - Copertura Requisiti Desiderabili	Funzionabilità
MPRS3 - Numero di Passaggi per Operazione	Usabilità
MPRS4 - Function Understandability	Usabilità
MPRS5 - Task Completion	Efficacia
MPRS6 - Numero di statement per metodo	Manutenibilità
MPRS7 - Numero di campi dati per classe	Manutenibilità
MPRS8 - Grado di accoppiamento	Manutenibilità
MPRS9 - Cyclomatic Number	Manutenibilità
MPRS10 - Conditional Statement	Manutenibilità
MPRS11 - Adequacy of variable names	Manutenibilità
MPRS12 - Average Module Size	Manutenibilità
MPRS13 - Test Passati Richiesti	Affidabilità
MPRS14 - Failure Avoidance	Affidabilità
MPRS15 - Breakdown Avoidance	Affidabilità
MPRS16 - Numero di installazioni completate	Installabilità

Tabella 1: Mappa Metriche-Caratteristiche

3.3.3.2.1 Copertura Requisiti Obbligatori - MPRS1 Questa metrica ci permette di verificare in ogni momento lo stato dei requisiti obbligatori coperti. Essa misura il rapporto tra requisiti obbligatori soddisfatti e il numero totale di requisiti obbligatori ricavati. Per una maggiore comprensione il risultato verrà riportato in percentuale.

$$\text{Copertura Requisiti Obbligatori} = \frac{\# \text{ requisiti obbligatori soddisfatti}}{\# \text{ requisiti obbligatori totali}} \quad (2)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 100% è ritenuta negativa;
- una percentuale uguale al 100% è ritenuta accettabile;
- una percentuale uguale al 100% ottimale.

3.3.3.2.2 Copertura Requisiti Desiderabili - MPRS2 Questa metrica ci permette di verificare in ogni momento lo stato dei requisiti coperti. Essa misura il rapporto tra requisiti desiderabili soddisfatti e il numero totale di requisiti desiderabili ricavati. Per una maggiore comprensione il risultato verrà riportato in percentuale.

$$\text{Copertura Requisiti Desiderabili} = \frac{\# \text{ requisiti desiderabili soddisfatti}}{\# \text{ requisiti desiderabili totali}} \quad (3)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 100% è ritenuta negativa;
- una percentuale uguale al 100% è ritenuta accettabile;
- una percentuale uguale al 100% ottimale.

3.3.3.2.3 Numero di Passaggi per Operazione - MPRS3 La metrica verifica che il numero di passaggi per effettuare un'operazione sia contenuto. Questa metrica assicura che il prodotto sia semplice da utilizzare e comporti poco sforzo all'utente nel compiere un'operazione.

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un numero di passaggi maggiore di 8 è ritenuto negativo;
- un numero di passaggi compreso tra 5 e 8 è ritenuto accettabile;
- un numero di passaggi minore di 5 è ritenuto ottimale.

3.3.3.2.4 Function Understandability - MPRS4 La metrica misura il rapporto tra le funzionalità capite dall'utente e le funzionalità offerte dal prodotto_g. Questo ci permette di capire se l'utente riesce a comprendere fin da subito le funzionalità offerte. Per una maggiore comprensione il risultato verrà riportato in percentuale.

$$Function\ Understandability = \frac{\# \text{ funzionalità capite}}{\# \text{ funzionalità disponibili}} \quad (4)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 70% è ritenuta negativa;
- una percentuale compresa tra 70% e 80% è ritenuta accettabile;
- una percentuale maggiore del 80% è ritenuta ottimale.

3.3.3.2.5 Task Completion - MPRS5 La metrica misura la percentuale media di task completati da un'utente. Questo ci permette di valutare l'efficacia con cui gli utenti possano portare a termine determinate azioni.

$$Task\ Completion = \frac{1}{N} \sum_1^N \frac{\# \text{ task completati}}{\# \text{ task assegnati}} N = \text{numero di utenti} \quad (5)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale del 70% è ritenuta negativa;
- una percentuale compresa tra 70% e 80% è ritenuta accettabile;
- una percentuale maggiore di 80% è ritenuta ottimale.

3.3.3.2.6 Numero di statement per metodo - MPRS6 La metrica misura il numero di statement per metodo. Questo ci permette di tenere un grado di manutenibilità del codice accettabile. Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore maggiore di 40 è ritenuto negativo;
- un valore compreso tra 30 e 40 è ritenuto accettabile;
- un valore minore di 30 è ritenuto ottimale.

3.3.3.2.7 Numero di campi dati per classe - MPRS7 La metrica misura il numero di campi dati per classe. Questo ci permette di tenere un grado di manutenibilità e comprensibilità del codice accettabile. Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore maggiore di 15 è ritenuto negativo;
- un valore tra 10 e 15 è ritenuto accettabile;
- un valore minore di 10 è ritenuto ottimale.

3.3.3.2.8 Grado di accoppiamento - MPRS8 La metrica misura il numero di dipendenze tra classi di un package_g. Avere poche dipendenze tra classi implica che ci sia un maggiore grado di disaccoppiamento. Questo aumenta molto la manutenibilità e la comprensibilità del codice. Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore maggiore di 7 è ritenuto negativo;
- un valore compreso tra 3 e 7 è ritenuto accettabile;
- un valore minore di 3 è ritenuto ottimale.

3.3.3.2.9 Cyclomatic Number - MPRS9 La metrica misura la complessità ciclomatica del codice. Essa ci permette di vedere il numero di cammini linearmente indipendenti presenti all'interno del codice. Il numero di cammini è direttamente proporzionale alla complessità ciclomatica del codice.

$$\text{Cyclomatic Number} = e - n + 2p \quad (6)$$

- e = numero di archi;
- n = numero di nodi;
- p = numero di componenti connesse.

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore maggiore di 10 è ritenuto negativo;
- un valore compreso tra 4 e 10 è ritenuto accettabile;
- un valore minore di 4 è ritenuto ottimale.

3.3.3.2.10 Conditional Statement - MPRS10 La metrica misura il numero di statement condizionali presenti per modulo. Questo ci permette di valutare la complessità del codice di un modulo. Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore maggiore di 50 è ritenuto negativo;
- un valore compreso tra 20 e 50 è ritenuto accettabile;
- un valore minore di 20 è ritenuto ottimale.

3.3.3.2.11 Adequacy of variable names - MPRS11 La metrica misura la divergenza dei nomi delle variabili da quanto deciso nella *Definizione di prodotto v1.00*. Questo ci permette di mantenere una buona comprensibilità del codice.

$$\text{Adequacy of variable names} = \frac{\# \text{ variabili con nomi corretti}}{\# \text{ totale di variabili definite nella DP}} \quad (7)$$

- DP = *Definizione di prodotto v1.00*.

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore minore di 80% è ritenuto negativo;
- un valore compreso tra 80% e 90% è ritenuto accettabile;
- un valore maggiore di 90% è ritenuto ottimale.

3.3.3.2.12 Average Module Size - MPRS12 La metrica misura la dimensione media di un modulo in termini di linee di codice. Avere moduli di grandi dimensioni compromette la manutenibilità del codice.

$$\text{Average module size} = \frac{\# \text{ totale di linee di codice in tutti i moduli}}{\# \text{ moduli}} \quad (8)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- un valore maggiore di 400 è ritenuto negativo;
- un valore compreso tra 300 e 400 è ritenuto accettabile;
- un valore compreso tra 200 e 300 è ritenuto ottimale.

3.3.3.2.13 Test Passati Richiesti - MPRS13 La metrica misura la percentuale di successo dei test ricavati dai requisiti. Questo ci permette di valutare se il prodotto supera la maggior parte dei test.

$$Test\ Passati\ Richiesti = \frac{\# \text{ di test passati}}{\# \text{ totale di test richiesti}} \quad (9)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 80% e maggiore del 98% è ritenuta negativa;
- una percentuale compresa tra 80% e 90% è ritenuta accettabile;
- una percentuale compresa tra 90% e 98% è ritenuta ottimale.

Da notare che una percentuale maggiore del 98% è ritenuta negativa, perché lo scopo dei test è quello di trovare dei problemi. Nel momento del raggiungimento del 100% dei test passati il set di test previsto verrà cambiato.

3.3.3.2.14 Failure Avoidance - MPRS14 La metrica misura la percentuale di situazioni anomale evitate dal prodotto in seguito ad un numero di situazioni anomale presentate. Questo ci permette di valutare se il prodotto è robusto e risponde bene ad eventuali situazioni anomale.

$$Failure\ Avoidance = \frac{\# \text{ situazioni anomale evitate}}{\# \text{ totale di situazioni anomale presentate}} \quad (10)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 80% è ritenuta negativa;
- una percentuale compresa tra 80% e 90% è ritenuta accettabile;
- una percentuale maggiore di 90% è ritenuta ottimale.

3.3.3.2.15 Breakdown Avoidance - MPRS15 La metrica misura la percentuale di interruzioni evitate dal prodotto in seguito ad un numero di situazioni anomale presentate. Questo ci permette di valutare se il prodotto lavora senza interruzioni.

$$Breakdown\ Avoidance = 1 - \frac{\# \text{ di interruzioni}}{\# \text{ di situazioni anomale presentate}} \quad (11)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 80% è ritenuta negativa;
- una percentuale compresa tra 80% e 90% è ritenuta accettabile;
- una percentuale maggiore di 90% è ritenuta ottimale.

3.3.3.2.16 Numero di installazioni completate - MPRS15 La metrica misura la percentuale di installazioni avvenute con successo da parte degli utenti. Questo ci permette di valutare se il prodotto è facilmente installabile.

$$\text{Numero di installazioni completate} = \frac{\# \text{ installazioni avvenute con successo}}{\# \text{ totale di installazioni}} \quad (12)$$

Vengono di seguito riportati i range stabiliti per la metrica appena introdotta:

- una percentuale minore del 80% è ritenuta negativa;
- una percentuale compresa tra 80% e 90% è ritenuta accettabile;
- una percentuale maggiore di 90% è ritenuta ottimale.

A Capability Maturity Model

Il CMM_g, acronimo di Capability Maturity Model, è un modello di sviluppo creato in seguito allo studio, finanziato dal Dipartimento della Difesa Statunitense, dei dati raccolti dalle organizzazioni che collaboravano con esso. Tale modello mira a migliorare i processi di sviluppo software_g esistenti. Il nome stesso del modello suggerisce i concetti su cui si basa:

capability: è una caratteristica di ogni processo_g che indica quanto esso sia adeguato per gli scopi per cui è stato definito; tale caratteristica determina l'apporto in termini di efficacia ed efficienza finali raggiungibile attraverso un processo_g;

maturity: è una caratteristica di un insieme di processi, attraverso la quale è possibile misurare quanto è governato il sistema dei processi di un'azienda;

model: è la definizione di un insieme di requisiti, sempre più stringenti, che consentono di valutare il percorso di miglioramento dei processi di un'azienda.

Il modello CMM_g fornisce:

- una base concettuale su cui appoggiarsi per valutare il livello dei processi;
- un insieme di best practice consolidate negli anni da esperti e utilizzatori;
- un linguaggio comune e una visione condivisa;
- un metodo per definire un miglioramento in ambito organizzativo.

A.1 Struttura

Il modello CMM_g comprende cinque aspetti:

Livelli di maturità: sono cinque livelli di maturità, dove il più alto (il quinto) è uno stato teoricamente ideale in cui i processi vengono sistematicamente gestiti attraverso una combinazione di ottimizzazioni e miglioramenti continui di processi;

Aree chiave di processo_g: un'area chiave di processo_g identifica un gruppo di attività correlate che, quando vengono eseguite insieme, producono una serie di obiettivi considerati strategici;

Obiettivi: gli obiettivi di un'area chiave di processo_g riassumono gli stati che devono esistere per quell'area per essere implementati in modo completo e duraturo. La quantità di obiettivi che sono stati raggiunti è un indicatore della capability che l'organizzazione ha raggiunto in un certo livello di maturità;

Caratteristiche comuni: le caratteristiche comuni includono le pratiche che sviluppano e regolamentano un'area chiave di processo_g. Ci sono cinque tipi di caratteristiche comuni:

- l'impegno nell'esecuzione;
- l'abilità nell'esecuzione;
- le attività eseguite;
- le misurazioni e le analisi;
- la verifica e l'implementazione.

Le pratiche fondamentali: le pratiche fondamentali descrivono gli elementi dell'infrastruttura e le pratiche che contribuiscono in modo particolare all'implementazione e alla regolamentazione dell'area.

A.2 Livelli

Il primo e più importante degli aspetti del modello visti nella sezione precedente riguarda i livelli che indicano il grado di maturità raggiunto da un'azienda.

Primo livello - Iniziale (Caotico) I processi che rientrano fra quelli di questo livello tipicamente risultano privi di ogni forma di documentazione e in uno stato di continuo cambiamento, riadattati ogni volta alle necessità del momento, poco riusabili e incontrollati. Tutto ciò porta ad un ambiente caotico e instabile per i processi.

Secondo livello - Ripetibile I processi di questo livello sono generalmente ripetibili, e spesso danno buoni risultati; inizia a vedersi una certa disciplina nei processi che li porta ad essere rigorosi e robusti.

Terzo livello - Definito I processi iniziano ad essere raggruppati secondo standard definiti, vengono documentati e sono soggetti a miglioramenti nel lungo periodo. A questo livello gli standard di processo_g sono usati per consolidare l'esecuzione dei processi nell'organizzazione.

Quarto livello - Gestito A questo livello iniziano ad essere usate metriche di processo, e i manager dell'azienda sono in grado di individuare i modi di adeguare e migliorare i processi rispetto a specifici progetti, senza rilevare perdite di qualità o deviazioni dalle specifiche.

Quinto livello - Ottimizzante I processi in questo livello sono volti a migliorare continuamente le performance attraverso cambiamenti e miglioramenti sia incrementali che tecnologicamente innovativi.

B Standard ISO/IEC 9126

Le norme ISO/IEC 9126 descrivono un modello di qualità del software, definiscono le caratteristiche che la determinano e propongono metriche per la misurazione.

Tali norme suddividono i criteri qualitativi in tre aree.

- **Qualità interna:** è la qualità del prodotto software vista dall'interno e fa quindi riferimento alle caratteristiche implementative del software quali l'architettura e il codice che ne deriva.
- **Qualità esterna:** è la qualità del prodotto software vista dall'esterno nel momento in cui esso viene eseguito e testato in un ambiente di prova.
- **Qualità in uso:** è la qualità del prodotto software dal punto di vista dell'utilizzatore che ne fa uso all'interno di uno specifico sistema e contesto.

Le norme constano di quattro parti:

1. modello della qualità del software;
2. metriche per la qualità interna;
3. metriche per la qualità esterna;
4. metriche per la qualità in uso.

B.1 Modello della qualità del software

Nella prima parte dello standard vengono descritti il modello relativo alla qualità esterna e interna e il modello riguardante la qualità in uso.

B.1.1 Modello della qualità interna ed esterna

Il modello di qualità esterna ed interna stabilito nella prima parte dello standard è classificato da sei caratteristiche generali:

funzionalità: La funzionalità rappresenta la capacità del software di fornire le funzioni necessarie per operare in determinate condizioni, cioè in un determinato contesto.

affidabilità: L'affidabilità rappresenta la capacità di un prodotto software di mantenere il livello di prestazione quando viene utilizzato in condizioni specificate e per un certo periodo di tempo.

usabilità: L'usabilità rappresenta la capacità di un prodotto software di essere comprensibile. Un software è considerato usabile in proporzione alla facilità con cui gli utenti operano per sfruttare a pieno le funzionalità che il software realizza.

efficienza: L'efficienza rappresenta la capacità di un prodotto di realizzare le funzioni richieste nel minor tempo possibile ed utilizzando nel miglior modo le risorse necessarie, quando opera in determinate condizioni.

manutenibilità: La manutenibilità rappresenta la capacità di un prodotto software di essere modificato (a costi accessibili ed in tempi rapidi). Le modifiche possono includere correzioni o adattamenti del software a modifiche negli ambienti, nei requisiti e nelle specifiche funzionali.

portabilità: La portabilità rappresenta la capacità di un prodotto software di poter essere trasportato da un ambiente all'altro (in modo sufficientemente veloce). L'ambiente include aspetti hardware e software.

Tali caratteristiche sono misurabili attraverso delle metriche.

B.1.2 Modello della qualità in uso

Gli attributi presenti nel modello relativo alla qualità del software in uso sono rappresentati da quattro grandi categorie:

efficacia: L'efficacia di un prodotto software rappresenta la capacità di permettere all'utente di raggiungere obiettivi specifici con accuratezza e completezza in uno specifico contesto di utilizzo.

produttività: La produttività di un prodotto software rappresenta la capacità di permettere all'utente di impegnare un numero definito di risorse, in relazione all'efficienza raggiunta in uno specifico contesto di utilizzo.

sicurezza fisica: La sicurezza fisica di un prodotto software rappresenta la capacità di raggiungere un livello accettabile di rischio per i dati, le persone, il business, la proprietà o gli ambienti in uno specifico contesto di utilizzo.

soddisfazione: La soddisfazione di un prodotto software rappresenta la capacità di soddisfare gli utenti in uno specifico contesto di utilizzo.

B.2 Metriche per la qualità interna

Le metriche interne si applicano al software non eseguibile (come, ad esempio, il codice sorgente) e alla documentazione. Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale in quanto, gli attributi interni influenzano le caratteristiche esterne e quelle in uso.

B.3 Metriche per la qualità esterna

Le metriche esterne misurano i comportamenti del prodotto software rilevabili dai test, dall'operatività, dall'osservazione durante la sua esecuzione. L'esecuzione del prodotto software è fatta in un contesto tecnico rilevante. Le metriche esterne sono scelte in base alle caratteristiche che il prodotto finale dovrà dimostrare durante la sua esecuzione in esercizio.

B.4 Metriche per la qualità in uso

Le metriche della qualità in uso misurano il grado con cui il prodotto software permette agli utenti di svolgere le proprie attività con efficacia, produttività, sicurezza e soddisfazione nel contesto operativo previsto.

C PDCA

Il PDCA, acronimo di Plan-Do-Check-Act, conosciuto anche come "Ciclo di Deming" o "Ciclo di miglioramento continuo", è un modello studiato per il miglioramento continuo della qualità in un'ottica a lungo raggio.

Questo strumento permette di fissare degli obiettivi di miglioramento a partire dagli esiti delle misurazioni effettuate durante le varie attività di verifica. Una volta fissati gli obiettivi che si desidera raggiungere, si iterano le attività previste dal Ciclo di Deming fino al raggiungimento di tali obiettivi.

I miglioramenti ai quali si fa riferimento sono legati all'efficienza e all'efficacia. Migliorare l'efficienza significa usare meno risorse per fare lo stesso lavoro. Migliorare l'efficacia significa divenire più conformi alle aspettative. Vengono riportate di seguito le quattro attività previste dal Ciclo di Deming:

- **Plan - Pianificare:** consiste nel definire gli obiettivi di miglioramento e le strategie da utilizzare per raggiungere tali obiettivi. Durante questa attività viene inoltre pianificato il modo in cui attuare queste strategie per raggiungere gli obiettivi di miglioramento fissati.
- **Do - Eseguire:** consiste nell'attuazione di quanto è stato pianificato al punto precedente. Oltre a fare ciò, si devono anche raccogliere i dati necessari all'analisi che viene svolta ai punti successivi.
- **Check - Verificare:** consiste nel verificare l'esito del processo (per efficienza ed efficacia) in seguito all'attuazione delle strategie di miglioramento. I risultati possono avere tre tipi di esito: si è migliorati nel modo atteso, si è fatto meglio o si è fatto peggio.
- **Act - Agire:** consiste nell'attuazione di soluzioni correttive, ovvero nell'attuazione delle strategie che hanno portato miglioramenti, anche al di fuori dei singoli processi per i quali si erano fissati gli obiettivi di miglioramento.

Si noti infine che se si vuole essere migliorabili bisogna essere analizzabili. E per essere analizzabili si deve essere ripetibili.

In altre parole: bisogna essere tracciabili, ovvero si devono eseguire le attività facendo sì che il modo in cui si ha operato possa essere valutato.

D Test

D.1 Test di accettazione

Il test di accettazione serve ad accertare il soddisfacimento dei **requisiti utente**. Viene effettuato in presenza del proponente che può, in questo modo, avere un primo approccio con il prodotto software, terminato. Nel caso in cui il test avesse esito positivo, si può procedere al rilascio ufficiale del prodotto, sviluppato.

Di seguito verranno riportati i test di accettazione, nel momento in cui il gruppo sarà in grado di definirli nel dettaglio.

D.2 Test di sistema

Il test di sistema verifica il comportamento dinamico del sistema completo al fine di verificare il soddisfacimento dei **requisiti software**. La maggior parte degli errori dovrebbe essere già stata identificata durante i test di unità e di integrazione. Il test di sistema viene di solito considerato appropriato per verificare il sistema anche rispetto ai requisiti non funzionali, come quelli prestazionali, di qualità e di vincolo. A questo livello, viene effettuata anche una serie di test in una struttura opportunamente mappata da beacon, per verificare il corretto funzionamento del software, ed evidenziare eventuali bug, o mancanze a livello di performance e precisione. Bisognerà verificare quindi:

- che il software, soddisfi tutti i requisiti obbligatori;
- che il software, soddisfi tutti i requisiti desiderabili ed opzionali che il gruppo si è impegnato a soddisfare;
- che il software, possieda tutte le caratteristiche garantite dal gruppo nel presente documento alla sezione 2.1.2 "Qualità di prodotto";
- come il software, si comporta su vari ambienti hardware che soddisfano i vincoli imposti nel documento *Analisi dei requisiti* alla sezione 2.5 "Vincoli generali";

Di seguito verranno riportati i test di sistema, nel momento in cui il gruppo sarà in grado di definirli nel dettaglio.

E Resoconto delle attività di verifica - Fase A

All'interno di questa prima fase_g, secondo quanto riportato nel documento *Piano di progetto v1.00*, sono previsti più momenti in cui viene attivato il processo_g di verifica. Si è cercato di riportare in questa sezione tutti i risultati che sono stati ottenuti durante questi momenti. Ove fosse necessario, si sono trattate anche delle conclusioni sui risultati ottenuti e su come essi possono essere migliorati.

E.1 Resoconto delle attività di verifica sui prodotti

In questa sezione verranno riportati i dati emessi dalle procedure di controllo della qualità di prodotto_g.

E.1.1 Documenti

In questa sezione vengono riportati i risultati delle attività di verifica svolte sui documenti. Esse sono di due tipi:

- verifiche manuali;
- verifiche automatizzate.

E.1.1.1 Verifiche manuali Le attività di verifica manuale della documentazione prodotta sono state svolte in base alla procedura riguardante la verifica dei documenti che è descritta nel documento *Norme di progetto v1.00*. La verifica manuale ha permesso di individuare soprattutto errori che riguardano le seguenti tipologie:

- periodi troppo lunghi e complessi da capire e interpretare;
- aggettivi o verbi utilizzati in modo non appropriato;
- incongruenze tra parti diverse dello stesso documento o appartenenti a documenti diversi;
- errori nei concetti esposti;
- violazioni di quanto stabilito nelle norme tipografiche.

Di seguito è presentato un riassunto della quantità di errori trovati (e successivamente risolti) utilizzando la verifica manuale durante l'intera fase_g A.

Periodi lunghi o complessi	11
Parole non appropriate	9
Incongruenze	15
Errori concettuali	18
Violazioni delle norme tipografiche	100

Tabella 2: Errori trovati tramite verifica manuale dei documenti durante la fase_g A

La verifica manuale, in aggiunta, ha permesso di individuare nuovi termini da aggiungere al *Glossario*. Di seguito è presentato un riassunto della quantità di nuovi termini da aggiungere al *Glossario* che sono stati individuati.

Termini candidati ad essere aggiunti	76
Termini aggiunti al <i>Glossario</i>	70

Tabella 3: Nuovi termini da inserire nel *Glossario* individuati tramite verifica manuale dei documenti durante la fase_g A

È stata infine verificata la correttezza dei diagrammi UML_g utilizzati all'interno dei vari documenti, sempre seguendo le procedure contenute nel documento *Norme di progetto v1.00*.

E.1.1.2 Verifiche automatiche Le attività di verifica automatizzate, oltre a rispettare le procedure descritte all'interno delle *Norme di progetto v1.00*, fanno uso degli strumenti automatici previsti all'interno dello stesso documento. Questi hanno permesso di individuare numerosi errori che riguardano le seguenti tipologie:

- ortografia errata;
- utilizzo errato dei comandi L^AT_EX previsti dalle *Norme di progetto v1.00*;
- norme tipografiche non rispettate (esempio: i punti e virgola necessari alla fine degli elenchi puntati o numerati, fatta eccezione per l'ultimo elemento dell'elenco che richiede il punto).

Di seguito è presentato un riassunto della quantità di errori trovati (e successivamente risolti) utilizzando la verifica automatica. Si tenga in considerazione il fatto che alcuni degli strumenti automatici utilizzati non sono stati disponibili fin dall'inizio.

Errori ortografici	31
Utilizzo errato L ^A T _E X	14

Tabella 4: Errori trovati tramite verifica automatica dei documenti durante la fase_g A

Merita un discorso a parte il calcolo dell'indice Gulpease_g, per il quale sono stati imposti nel presente documento dei range che determinano se un documento è accettabile o meno. Di seguito sono stati riportati gli indici ottenuti (relativi ai documenti completi).

Documento	Gulpease	Esito
<i>Piano di Progetto v1.00</i>	54	Ottimale
<i>Norme di Progetto v1.00</i>	60	Ottimale
<i>Studio di Fattibilità v1.00</i>	55	Ottimale
<i>Analisi dei Requisiti v1.00</i>	50	Ottimale
<i>Piano di Qualifica v1.00</i>	51	Ottimale
<i>Glossario v1.00</i>	67	Ottimale

Tabella 5: Esiti del calcolo dell'indice di leggibilità effettuato tramite strumenti automatici durante la fase_g A

E.2 Resoconto delle attività di verifica sui processi

E.2.1 Processo di documentazione

E.2.1.1 Livello CMM Il gruppo ha cercato di valutare la qualità del processo_g di documentazione secondo le metriche stabilite dal modello CMM_g: chiaramente, all'inizio della fase_g A il processo_g si posizionava al livello 1. In seguito alla redazione del documento *Norme di progetto* (uno dei primi ad essere realizzato) si ha avuto a disposizione norme valide per ogni tipo di documentazione, strumenti comuni da poter utilizzare e procedure da

seguire per effettuare determinate attività: questo ha permesso di controllare maggiormente il processo_g di documentazione, che ha in questo modo guadagnato ripetibilità (richiesta dal livello 2 di CMM_g). Quindi, possiamo affermare di aver raggiunto il livello 2 della scala CMM_g, perché il processo_g di documentazione non possiede ancora la principale caratteristica richiesta dal terzo livello, ovvero la proattività. Questo livello è ritenuto accettabile (secondo quanto descritto nel presente documento alla sezione 3.4 "Misure e metriche"), ma durante le prossime fasi si prevede comunque di continuare a lavorare per poter ottenere miglioramenti sotto questi punti di vista (sfruttando PDCA_g).

E.2.2 Processo di verifica

E.2.2.1 Livello CMM Essendo il processo_g di verifica molto costoso, il nostro obiettivo è di renderlo il più efficace e allo stesso tempo il più efficiente possibile. Per ottenere ciò, si deve rendere il processo_g controllabile.

Anche per quanto riguarda questo processo_g, come per quello di documentazione, siamo in grado di dire che è stato raggiunto il livello 2 nella scala prevista da CMM_g. Il processo_g ha infatti superato l'iniziale stato caotico nel quale si trovava all'inizio della fase_g A (grazie, per esempio, all'utilizzo sistematico di script e di procedure).

Il team_g non può ancora affermare che il processo_g di verifica adottato abbia raggiunto il livello 3 della scala CMM_g, in quanto è stata documentata in modo accettabile solo l'attività di realizzazione del processo_g, e non quella di gestione dello stesso; tuttavia il livello raggiunto è ritenuto accettabile (secondo quanto descritto nel presente documento alla sezione 3.4 "Misure e metriche"), anche se durante le prossime fasi si prevede comunque di continuare a lavorare per poter ottenere miglioramenti sotto questi punti di vista (sfruttando PDCA_g).