

CLIPS

Communication & Localization with Indoor Positioning Systems

UNIVERSITÀ DI PADOVA

PIANO DI QUALIFICA V4.00



leaf.gruppo@gmail.com

Versione	4.00
Data Redazione	2016-04-10
Redazione	Davide Castello
Verifica	Cristian Andrighetto
Approvazione	Oscar Elia Conti
Uso	Esterno
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo Miriade S.p.A.

Diario delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
4.00	2016-04-10	Oscar Elia Conti	Responsabile di progetto	Approvazione del documento
3.03	2016-04-10	Cristian Andrichetto	Verificatore	Verifica del documento
3.02	2016-04-09	Davide Castello	Amministratore	Aggiunto resoconto delle attività di verifica Fase PDROB
3.01	2016-04-08	Federico Tavella	Programmatore	Aggiunte sottosezioni test di unità e integrazione
3.00	2016-03-13	Marco Zanella	Responsabile di progetto	Approvazione del documento
2.02	2016-03-13	Davide Castello	Verificatore	Verifica del documento
2.01	2016-02-13	Federico Tavella	Amministratore	Resoconto delle attività di verifica - Fase PA
2.00	2016-02-22	Cristian Andrichetto	Responsabile di Progetto	Approvazione del documento
1.14	2016-02-21	Oscar Elia Conti	Verificatore	Verifica del documento
1.13	2016-02-21	Andrea Tombolato	Verificatore	Verifica del documento
1.12	2016-02-21	Davide Castello	Amministratore	Creato Resoconto delle attività di verifica - Fase AD
1.11	2016-02-20	Davide Castello	Amministratore	Rimossa sezione 3.2
1.10	2016-02-20	Davide Castello	Amministratore	Creazione appendice B e C

Versione	Data	Autore	Ruolo	Descrizione
1.09	2016-02-20	Cristian Andrighetto	Amministratore	Modifica paragrafo 2.1.2.2 - Aggiunta sottoparagrafi da 2.2.1.2.1 a 2.2.1.2.9
1.08	2016-02-19	Davide Castello	Amministratore	Sistematizzazione delle metriche per i documenti
1.07	2016-02-19	Davide Castello	Amministratore	Aggiunte metriche dei processi Schedule Variance e Budget Variance
1.06	2016-02-18	Davide Castello	Amministratore	Resi quantitativi e sistemati gli obiettivi di qualità su processi e documenti
1.05	2016-02-17	Davide Castello	Amministratore	Aggiunte label alle metriche per riferimento all'interno dei rispettivi obiettivi di qualità
1.04	2016-02-17	Davide Castello	Amministratore	Numerati gli obiettivi di qualità e le metriche
1.03	2016-02-16	Cristian Andrighetto	Amministratore	Eliminata sezione 3.3
1.02	2016-02-16	Cristian Andrighetto	Amministratore	Eliminata sezione 2.3

Versione	Data	Autore	Ruolo	Descrizione
1.01	2016-02-16	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
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

Versione	Data	Autore	Ruolo	Descrizione
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	Cristian Andrighetto	Amministratore	Stesura tecniche di controllo della qualità
0.03	2015-12-11	Cristian Andrighetto	Amministratore	Stesura sezione riguardante le risorse
0.02	2015-12-08	Davide Castello	Amministratore	Stesura visione generale della strategia di gestione della qualità
0.01	2015-12-07	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	5
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	Manutenibilità e Comprensibilità del codice - OBPRS3	8
2.1.2.2.4	Copertura dei test richiesti - OBPRS4	10
2.1.2.2.5	Robustezza - OBPRS5	10
2.1.2.2.6	Funzionamento senza interruzioni - OBPRS6	11
2.2	Scadenze temporali	11
3	La strategia di gestione della qualità nel dettaglio	12
3.1	Risorse	12
3.1.1	Risorse necessarie	12
3.1.1.1	Risorse umane	12
3.1.1.2	Risorse hardware	12
3.1.1.3	Risorse software	12
3.1.2	Risorse disponibili	12
3.1.2.1	Risorse umane	12

3.1.2.2	Risorse hardware	13
3.1.2.3	Risorse software	13
3.2	Misure e metriche	13
3.2.1	Misure	13
3.2.2	Metriche per i processi	13
3.2.2.1	Capability Maturity Model - MPC1	13
3.2.2.2	Schedule Variance - MPC2	14
3.2.2.3	Budget Variance - MPC3	14
3.2.3	Metriche per i prodotti	15
3.2.3.1	Metriche per i documenti	15
3.2.3.1.1	Indice di leggibilità - MPRD1	15
3.2.3.1.2	Errori ortografici rinvenuti e non corretti - MPRD2	16
3.2.3.1.3	Errori concettuali rinvenuti e non corretti - MPRD3	16
3.2.3.2	Metriche per il software	17
3.2.3.2.1	Copertura Requisiti Obbligatori - MPRS1	17
3.2.3.2.2	Copertura Requisiti Desiderabili - MPRS2	18
3.2.3.2.3	Numero di statement per metodo - MPRS3	18
3.2.3.2.4	Numero di parametri per metodo - MPRS4	18
3.2.3.2.5	Numero di campi dati per classe - MPRS5	19
3.2.3.2.6	Grado di accoppiamento - MPRS6	19
3.2.3.2.7	Cyclomatic Number - MPRS7	19
3.2.3.2.8	Adequacy of variable names - MPRS8	20
3.2.3.2.9	Average Module Size - MPRS9	20
3.2.3.2.10	Test Passati Richiesti - MPRS10	20
3.2.3.2.11	Failure Avoidance - MPRS11	21
3.2.3.2.12	Breakdown Avoidance - MPRS12	21
A	Capability Maturity Model	22
A.1	Struttura	22
A.2	Livelli	23
B	Standard ISO/IEC 9126	25
B.1	Modello della qualità del software	25
B.1.1	Modello della qualità esterna ed interna	25
B.1.2	Modello della qualità in uso	26
B.2	Metriche per la qualità del software	26

B.2.1	Metriche per la qualità esterna	27
B.2.2	Metriche per la qualità interna	27
B.2.3	Metriche per la qualità in uso	27
C	PDCA	28
D	Test	29
D.1	Test di accettazione	29
D.2	Test di sistema	62
D.3	Test di integrazione	74
D.3.1	Test-componenti	74
D.3.2	Componente - test	79
D.4	Test di unità	80
E	Resoconto delle attività di verifica - fase A	121
E.1	Resoconto delle attività di verifica sui prodotti	121
E.1.1	Documenti	121
E.1.1.1	Verifiche manuali	121
E.1.1.2	Verifiche automatiche	122
E.2	Resoconto delle attività di verifica sui processi	123
E.2.1	processo di documentazione	123
E.2.1.1	Livello CMM	123
E.2.2	processo di verifica	124
E.2.2.1	Livello CMM	124
F	Resoconto delle attività di verifica - fase AD	125
F.1	Verifica sui processi	125
F.1.1	processo di documentazione	125
F.1.1.1	Miglioramento costante	125
F.1.1.2	Rispetto della pianificazione	125
F.1.1.3	Rispetto del budget	126
F.1.2	processo di verifica	127
F.1.2.1	Miglioramento costante	127
F.1.2.2	Rispetto della pianificazione	127
F.1.2.3	Rispetto del budget	127
F.2	Verifica sui prodotti	128
F.2.1	Documenti	128
F.2.1.1	Leggibilità e comprensibilità	128
F.2.1.2	Correttezza ortografica	128
F.2.1.3	Correttezza concettuale	128

G	Resoconto delle attività di verifica - fase PA	130
G.1	Verifica sui processi	130
G.1.1	processo di documentazione	130
G.1.1.1	Miglioramento costante	130
G.1.1.2	Rispetto della pianificazione	130
G.1.1.3	Rispetto del budget	131
G.1.2	processo di verifica	131
G.1.2.1	Miglioramento costante	131
G.1.2.2	Rispetto della pianificazione	132
G.1.2.3	Rispetto del budget	132
G.2	Verifica sui prodotti	132
G.2.1	Documenti	132
G.2.1.1	Leggibilità e comprensibilità	133
G.2.1.2	Correttezza ortografica	133
G.2.1.3	Correttezza concettuale	133
H	Resoconto delle attività di verifica - fase PDROB	135
H.1	Verifica sui processi	135
H.1.1	processo di documentazione	135
H.1.1.1	Miglioramento costante	135
H.1.1.2	Rispetto della pianificazione	135
H.1.1.3	Rispetto del budget	136
H.1.2	processo di verifica	136
H.1.2.1	Miglioramento costante	136
H.1.2.2	Rispetto della pianificazione	137
H.1.2.3	Rispetto del budget	137
H.2	Verifica sui prodotti	137
H.2.1	Documenti	137
H.2.1.1	Leggibilità e comprensibilità	138
H.2.1.2	Correttezza ortografica	138
H.2.1.3	Correttezza concettuale	138

Elenco delle tabelle

1	Mappa Metriche-Caratteristiche	17
2	Tabella test di accettazione	61
3	Tabella di tracciamento test di sistema / requisiti	74
4	Tabella test di integrazione	79
5	Tabella componente / test di integrazione	79
6	Tabella descrizione test unità	120
7	Errori trovati tramite verifica manuale dei documenti durante la fase A	122
8	Nuovi termini da inserire nel <i>Glossario</i> individuati tramite verifica manuale dei documenti durante la fase A	122
9	Errori trovati tramite verifica automatica dei documenti durante la fase A	122
10	Esiti del calcolo dell'indice di leggibilità effettuato tramite strumenti automatici durante la fase A	123
11	Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase AD	126
12	Esiti del calcolo della Budget Variance sul processo di documentazione durante la fase AD	126
13	Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase AD	127
14	Esiti del calcolo della Budget Variance sul processo di verifica durante la fase AD	127
15	Errori ortografici trovati tramite verifica automatica dei documenti durante la fase AD	128
16	Errori concettuali trovati tramite verifica manuale dei documenti durante la fase AD	129
17	Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase PA	130
18	Esiti del calcolo della Budget Variance sul processo di documentazione durante la fase PA	131
19	Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase PA	132
20	Esiti del calcolo della Budget Variance sul processo di verifica durante la fase PA	132
21	Errori ortografici trovati tramite verifica automatica dei documenti durante la fase PA	133
22	Errori concettuali trovati tramite verifica manuale dei documenti durante la fase PA	133

23	Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase PDROB	135
24	Esiti del calcolo della Budget Variance sul processo di documentazione durante la fase PDROB	136
25	Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase PDROB	137
26	Esiti del calcolo della Budget Variance sul processo di verifica durante la fase PDROB	137
27	Errori ortografici trovati tramite verifica automatica dei documenti durante la fase PDROB	138
28	Errori concettuali trovati tramite verifica manuale dei documenti durante la fase PDROB	138

1 Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo 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 Bluetooth_g Low Energy (BLE_g). Il prodotto_g comprenderà un prototipo software_g che permetta la navigazione all'interno di un'area predefinita, basandosi sui concetti di Indoor Positioning System (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 v4.00* 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 v4.00*;
- **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 v4.00*;

- 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: per ogni criterio utilizzato abbiamo fissato dei valori minimi che intendiamo raggiungere nell'arco dell'intero progetto. Oltre a ciò sono stati fissati dei valori ottimali che dovrebbero essere sperabilmente (ma non obbligatoriamente) raggiunti.

Inoltre, gli obiettivi riportati in questa sezione sono identificati da un codice identificativo, al fine di rendere più semplice il tracciamento tra l'obiettivo di qualità da soddisfare e la metrica che permette di verificarne il soddisfacimento.

La procedura di denominazione degli obiettivi è spiegata in dettaglio nel documento *Norme di progetto v4.00*.

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.

Desideriamo che i processi posseggano le seguenti caratteristiche ottimali:

- un processo_g dovrebbe essere in grado di migliorare continuamente le proprie performance
 - le performance di un processo_g dovrebbero essere costantemente misurabili;
 - un processo_g dovrebbe cercare di perseguire sempre degli obiettivi quantitativi di miglioramento.

- le attività di un processo, dovrebbero rispettare i tempi indicati nel documento *Piano di progetto v4.00*;
- i costi effettivi di ogni processo, dovrebbero essere in linea con quanto dichiarato nel documento *Piano di progetto v4.00*.

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 Si è deciso di adottare il modello CMM_g per quantificare la capacità dei processi di misurare le proprie performance e di porsi obiettivi quantitativi di miglioramento.

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_g;

Soglia di accettabilità: livello 2 previsto da CMM_g;

Soglia di ottimalità: livello 4 previsto da CMM_g;

Per una migliore e più dettagliata descrizione del modello CMM_g 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 [3.2.2.1](#).

2.1.1.2 Rispetto della pianificazione - OQPC2 Per capire se le attività di un processo, sono 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 [3.2.2.2](#).

2.1.1.3 Rispetto del budget - OQPC3 Per capire se i costi di un processo_g rientrano 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 [3.2.2.3](#).

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. Per mantenere la qualità del prodotto_g, il gruppo cercherà di seguire al meglio lo standard di qualità [ISO/IEC 9126:2001].

Il progetto prevede la produzione di due tipologie di prodotti: documenti e software_g; nelle prossime sezioni sono esposti gli obiettivi che il gruppo *Leaf* intende raggiungere, suddivisi per tipologia di prodotto_g.

Per ogni obiettivo, 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 e i valori ottimali che si intendono raggiungere.

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 [3.2.3.1.1](#).

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 [3.2.3.1.2](#).

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% (soglia di accettabilità) degli errori concettuali rinvenuti non siano corretti. L'obiettivo ottimale sarebbe quello di correggere tutti gli errori trovati.

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 [3.2.3.1.3](#).

2.1.2.2 Qualità del software Gli obiettivi di qualità del software_g ai quali il gruppo *Leaf* desidera arrivare nell'arco del progetto sono un sottoinsieme di quelli enunciati nello standard [ISO/IEC 9126:2001]:

- 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 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_g;
- il prodotto_g garantisce un funzionamento senza interruzioni.

2.1.2.2.1 Funzionalità obbligatorie - OBPRS1 Il prodotto_g 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 [3.2.3.2.1](#).

2.1.2.2.2 Funzionalità desiderabili - OBPRS2 Il prodotto_g 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 [3.2.3.2.2](#).

2.1.2.2.3 Manutenibilità e Comprensibilità del codice - OBPRS3 Il prodotto_g deve avere codice manutenibile e non deve generare 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_g.

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 parametri per metodo

Soglia di accettabilità: $4 < X < 10$

Soglia di ottimalità: $X < 4$

X = numero di parametri 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 ciclomatica

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: Average Module Size

Soglia di accettabilità: $300 < X < 400$

Soglia di ottimalità: $200 < X < 300$

X = numero di linee di codice per modulo_g

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

- [3.2.3.2.3](#) - MPRS3 - Numero di statement per metodo
- [3.2.3.2.4](#) - MPRS4 - Numero di parametri per metodo
- [3.2.3.2.5](#) - MPRS5 - Numero di campi dati per classe
- [3.2.3.2.6](#) - MPRS6 - Grado di accoppiamento
- [3.2.3.2.7](#) - MPRS7 - Cyclomatic Number
- [3.2.3.2.8](#) - MPRS8 - Adequacy of variable names
- [3.2.3.2.9](#) - MPRS9 - Average Module Size

2.1.2.2.4 Copertura dei test richiesti - OBPRS4 Il prodotto_g deve essere testato in ogni sua parte per garantirne il 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 MPRS10 "Test Passati Richiesti" nella sezione [3.2.3.2.10](#).

2.1.2.2.5 Robustezza - OBPRS5 Il prodotto_g deve essere robusto e non deve interrompere il suo funzionamento in seguito al verificarsi di situazioni anomale_g. Il prodotto_g 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_g evitate su situazioni anomale_g 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 MPRS11 "Failure Avoidance" nella sezione [3.2.3.2.11](#).

2.1.2.2.6 Funzionamento senza interruzioni - OBPRS6 Il prodotto_g deve garantire un funzionamento senza interruzioni. Questo livello è considerato ottimale ma secondo la metrica scelta possono esserci al massimo il 20% di interruzioni dovute al verificarsi di situazioni anomale_g.

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_g 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 MPRS12 "Breakdown Avoidance" nella sezione [3.2.3.2.12](#).

2.2 Scadenze temporali

Le scadenze che il gruppo *Leaf* ha deciso di rispettare sono riportate nel *Piano di progetto v4.00*.

3 La strategia di gestione della qualità nel dettaglio

3.1 Risorse

Per garantire un buon funzionamento del processo_g di verifica verranno impiegati i seguenti tipi di risorse:

- 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 v4.00*.

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 v4.00*.

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(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 Tutti i membri del gruppo sono a disposizione per eseguire operazioni di verifica. Ognuno dei componenti, a turno, ricoprirà il ruolo di *Responsabile di progetto* o di *Verificatore* come definito nel *Piano di progetto v4.00*.

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 comprendono editor L^AT_EX 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 v4.00*.

3.2 Misure e metriche

3.2.1 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.

3.2.2 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 v4.00*.

3.2.2.1 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.

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

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

3.2.3 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 v4.00*.

3.2.3.1 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.

3.2.3.1.1 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.

3.2.3.1.2 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.

3.2.3.1.3 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;

3.2.3.2 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 Obbligatoriosi	Funzionalità
MPRS2 - Copertura Requisiti Desiderabili	Funzionalità
MPRS3 - Numero di statement per metodo	Manutenibilità
MPRS4 - Numero di parametri per metodo	Manutenibilità
MPRS5 - Numero di campi dati per classe	Manutenibilità
MPRS6 - Grado di accoppiamento	Manutenibilità
MPRS7 - Cyclomatic Number	Manutenibilità
MPRS8 - Adequacy of variable names	Manutenibilità
MPRS9 - Average Module Size	Manutenibilità
MPRS10 - Test Passati Richiesti	Affidabilità
MPRS11 - Failure Avoidance	Affidabilità
MPRS12 - Breakdown Avoidance	Affidabilità

Tabella 1: Mappa Metriche-Caratteristiche

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

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

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.

3.2.3.2.2 Copertura Requisiti Desiderabili - MPRS2 Questa metrica ci permette di verificare in ogni momento lo stato dei requisiti desiderabili coperti. Essa controlla il rapporto tra i 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.

3.2.3.2.3 Numero di statement per metodo - MPRS3 La metrica è in grado di determinare se il numero di statement per metodo implementati rientri tra i valori definiti. Questo ci permette di tenere un livello 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.

3.2.3.2.4 Numero di parametri per metodo - MPRS4 La metrica è in grado di determinare se il numero di parametri di un metodo rientri tra i valori definiti. 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 10 è ritenuto negativo;
- un valore compreso tra 4 e 10 è ritenuto accettabile;
- un valore minore di 4 è ritenuto ottimale.

3.2.3.2.5 Numero di campi dati per classe - MPRS5 La metrica permette di verificare che il numero di campi dati per classe rientri tra i valori definiti. 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.

3.2.3.2.6 Grado di accoppiamento - MPRS6 La metrica determina se il numero di dipendenze tra classi in un package, rientri tra i valori definiti. 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.

3.2.3.2.7 Cyclomatic Number - MPRS7 La metrica controlla che la complessità ciclomatica del codice rientri tra i valori definiti. 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.

$$\text{Cyclomatic Number} = e - n + 2p \quad (4)$$

- 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.

3.2.3.2.8 Adequacy of variable names - MPRS8 La metrica controlla che la divergenza dei nomi delle variabili da quanto deciso nella *Definizione di prodotto v1.00* rientri tra i valori definiti. Questo ci permette di mantenere una buona comprensibilità del codice.

$$\text{Adequacy of variable names} = \frac{\# \text{ variabili con nomi corretti}}{\# \text{ totale di variabili definite nella DP}} \quad (5)$$

- 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.

3.2.3.2.9 Average Module Size - MPRS9 La metrica controlla che la dimensione media di un modulo_g in termini di linee di codice rientri tra i valori definiti. Avere moduli di grandi dimensioni compromette la manutenibilità del codice.

$$\text{Average module size} = \frac{\# \text{ totale di linee di codice in tutti i moduli}}{\# \text{ moduli}} \quad (6)$$

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.

3.2.3.2.10 Test Passati Richiesti - MPRS10 La metrica controlla che la percentuale di successo dei test ricavati dai requisiti rientri tra i valori definiti. Questo ci permette di valutare se il prodotto_g supera la maggior parte dei test.

$$\text{Test Passati Richiesti} = \frac{\# \text{ di test passati}}{\# \text{ totale di test richiesti}} \quad (7)$$

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, perchè lo scopo dei test è quello di trovare dei problemi. Al raggiungimento del 100% dei test passati il set di test previsto verrà cambiato.

3.2.3.2.11 Failure Avoidance - MPRS11 La metrica controlla che la percentuale di situazioni anomale_g evitate dal prodotto_g rientri nei valori definiti. Questo ci permette di valutare se il prodotto_g è robusto e risponde bene ad eventuali situazioni anomale_g.

$$Failure\ Avoidance = \frac{\# \text{ situazioni anomale evitate}}{\# \text{ totale di situazioni anomale presentate}} \quad (8)$$

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.

3.2.3.2.12 Breakdown Avoidance - MPRS12 La metrica controlla che la percentuale di interruzioni evitate dal prodotto_g rientri tra i valori definiti. Il valore su cui si applicherà la metrica verrà calcolato come il complemento delle interruzioni verificate. Questa metrica ci permette di controllare che il prodotto_g lavori senza interruzioni.

$$Breakdown\ Avoidance = 1 - \frac{\# \text{ di interruzioni}}{\# \text{ di situazioni anomale presentate}} \quad (9)$$

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

La creazione del CMM_g, acronimo di Capability Maturity Model, è stata voluta e finanziata dal Dipartimento della Difesa Statunitense per valutare la qualità dei processi delle 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 che misura la convergenza di un processo_g rispetto agli scopi per cui è stato definito;

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, 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, 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. 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 in questo livello tipicamente risultano privi di ogni forma di documentazione e in uno stato di continuo cambiamento. Ad esempio vengono riadattati ogni volta alle necessità del momento risultando 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, 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

Lo Standard ISO/IEC 9126 si suddivide in quattro parti:

1. modello della qualità del software_g (9126-1);
2. metriche per la qualità esterna (9126-2);
3. metriche per la qualità interna (9126-3);
4. metriche per la qualità in uso (9126-4).

Lo standard tratta la qualità del software_g da tre punti di vista:

qualità interna: è la qualità del prodotto_g software_g vista dall'interno e fa quindi riferimento alle caratteristiche implementative del software_g quali l'architettura e il codice che ne deriva.

qualità esterna: è la qualità del prodotto_g software_g 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_g software_g dal punto di vista dell'utilizzatore che ne fa uso all'interno di uno specifico sistema e contesto.

B.1 Modello della qualità del software

Nella prima parte vengono descritti i modelli per la qualità esterna, interna ed in uso.

B.1.1 Modello della qualità esterna ed interna

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_g di fornire le funzioni necessarie per operare in determinate condizioni, cioè in un determinato contesto.

affidabilità: l'affidabilità è la capacità del prodotto_g software_g di mantenere un certo livello di prestazioni quando viene usato in condizioni specifiche e per un intervallo di tempo fissato.

usabilità: l'usabilità rappresenta la capacità di un prodotto_g software_g di essere comprensibile. Un software_g 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_g di realizzare le funzioni richieste nel minor tempo possibile ed utilizzando nel miglior modo le risorse necessarie.

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

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

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_g in uso sono rappresentati da quattro grandi categorie:

efficacia: l'efficacia di un prodotto_g software_g 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_g software_g 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_g software_g 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_g software_g rappresenta la capacità di soddisfare gli utenti in uno specifico contesto di utilizzo.

B.2 Metriche per la qualità del software

Nelle restanti tre parti vengono trattate le metriche per la qualità esterna, interna e in uso.

B.2.1 Metriche per la qualità esterna

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

B.2.2 Metriche per la qualità interna

Le metriche interne si applicano al software_g 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_g finale poiché gli attributi interni influenzano le caratteristiche esterne e quelle in uso.

B.2.3 Metriche per la qualità in uso

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

C PDCA

Il PDCA_g, 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 desiderano 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_g (per efficienza ed efficacia) in seguito all'attuazione delle strategie di miglioramento. I risultati possono avere tre tipi di esito:
 - un miglioramento secondo le aspettative;
 - un miglioramento superiore alle aspettative;
 - un miglioramento inferiore alle aspettative.
- **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 stati fissati gli obiettivi di miglioramento.

Bisogna tener presente che se l'obiettivo è il miglioramento continuo, le attività devono essere analizzabili, ripetibili e tracciabili. Unendo queste tre caratteristiche è possibile individuare eventuali errori e correggerli.

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 vengono riportati i test di accettazione definiti dal gruppo *Leaf*.

Codice	Requisito	Descrizione	Stato
TA1	L'utente deve poter verificare che sia possibile navigare all'interno di un edificio utilizzando l'applicazione.	All'utente è chiesto di: <ul style="list-style-type: none">● attivare il bluetooth;● accedere alla sezione preposta alla navigazione;● scegliere la destinazione all'interno dell'edificio;● confermare la destinazione scelta;● verificare che venga data la possibilità di avviare la navigazione.	N.I.
TA1.1	L'utente deve poter verificare che sia possibile ricercare una destinazione per la navigazione.	All'utente è chiesto di: <ul style="list-style-type: none">● scegliere di ricercare la destinazione per nome;● scegliere di ricercare la destinazione per categoria.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.1.1	L'utente deve poter verificare che sia possibile ricercare una destinazione per nome.	All'utente è chiesto di: <ul style="list-style-type: none">● inserire il nome di una destinazione;● verificare che sia possibile confermare l'inserimento fatto.	N.I.
TA1.1.1.1	L'utente deve poter verificare che sia possibile inserire il nome di una destinazione.	All'utente è chiesto di: <ul style="list-style-type: none">● inserire il nome di una destinazione;● verificare che la destinazione voluta sia stata inserita.	N.I.
TA1.1.1.2	L'utente deve poter verificare che venga segnalato un errore qualora venga inserita una destinazione non prevista dal sistema.	All'utente è chiesto di: <ul style="list-style-type: none">● inserire una destinazione non prevista dal sistema;● verificare che venga visualizzato un errore che spieghi che la destinazione inserita non è presente tra quelle disponibili;● verificare che venga data la possibilità di inserire un'altra destinazione.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.1.2	L'utente deve poter verificare che sia possibile ricercare una destinazione per categoria.	All'utente è chiesto di: <ul style="list-style-type: none">● scegliere una delle categorie proposte● scegliere una delle destinazioni presenti all'interno della categoria scelta;● verificare che sia possibile confermare la scelta fatta.	N.I.
TA1.1.2.1	L'utente deve verificare che sia possibile scegliere una categoria tra quelle proposte.	All'utente è chiesto di: <ul style="list-style-type: none">● verificare che l'applicazione fornisca una o più categorie di destinazioni;● verificare che sia possibile scegliere una di queste categorie.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.1.2.2	L'utente deve verificare che sia possibile scegliere una destinazione tra i risultati di una ricerca.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che all'interno di una categoria siano proposte una o più destinazioni;• verificare che sia possibile scegliere una di queste destinazioni;• effettuare la ricerca di una destinazione (prevista dal sistema) per nome;• verificare che sia possibile scegliere una delle destinazioni restituite dalla ricerca.	N.I.
TA1.1.3	L'utente deve poter verificare che sia possibile confermare una destinazione.	All'utente è chiesto di: <ul style="list-style-type: none">• confermare la destinazione scelta;• verificare che venga data la possibilità di avviare la navigazione verso la destinazione scelta.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.2	L'utente deve poter verificare che sia possibile avviare la navigazione.	All'utente è chiesto di: <ul style="list-style-type: none">• confermare l'avvio della navigazione;• verificare che venga fornita un'indicazione testuale per raggiungere la destinazione scelta.	N.I.
TA1.2.1	L'utente deve poter verificare che l'indicazione sia fornita in forma testuale.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che l'indicazione fornita sia un testo scritto.	N.I.
TA1.2.2	L'utente deve poter verificare che l'indicazione testuale fornita per raggiungere la destinazione scelta, quando è stata avviata la navigazione, sia corretta.	All'utente è chiesto di: <ul style="list-style-type: none">• seguire l'indicazione testuale data;• verificare di essere arrivato alla destinazione scelta.	N.I.
TA1.2.3	L'utente deve poter verificare che l'indicazione testuale fornita dall'applicazione abbia come punto di partenza il POI in cui l'utente si trova.	All'utente è chiesto di: <ul style="list-style-type: none">• avviare la navigazione;• verificare che l'indicazione testuale fornita dall'applicazione abbia come punto di partenza il POI in cui l'utente si trova.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.3	L'utente deve verificare che sia possibile interrompere la navigazione.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • scegliere di interrompere la navigazione; • verificare che la navigazione si interrompa. 	N.I.
TA1.4	L'utente deve verificare che sia possibile accedere a delle informazioni più dettagliate riguardanti il percorso da seguire per raggiungere la destinazione.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • scegliere di visualizzare le fotografie della prossima area; • scegliere di ricevere delle indicazioni testuali estese per raggiungere la prossima area; • scegliere di accedere alla lista completa delle indicazioni per raggiungere la destinazione scelta. 	N.I.
TA1.4.1	L'utente deve verificare che sia possibile visualizzare le fotografie della prossima area da raggiungere.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che siano visualizzate le fotografie che ritraggono la prossima area da raggiungere. 	N.I.

Codice	Requisito	Descrizione	Stato
TA1.4.2	L'utente deve verificare che sia possibile visualizzare delle indicazioni testuali estese dettagliate riguardanti le azioni da compiere per raggiungere la prossima area.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia visualizzata una descrizione testuale estesa che descriva in modo dettagliato le azioni da compiere per raggiungere la prossima area. 	N.I.
TA1.4.3	L'utente deve verificare che sia possibile visualizzare la lista completa delle indicazioni da seguire per raggiungere la destinazione scelta.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia visualizzata la lista completa delle indicazioni da seguire per raggiungere la destinazione scelta. 	N.I.
TA1.4.4	L'utente deve verificare che venga visualizzato un errore nel caso in cui acceda alla foto del prossimo POI con connessione Internet non attiva.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • disattivare la connessione Internet • accedere alle fotografie del prossimo POI • verificare che venga segnalato un errore che spieghi che il dispositivo non ha la connessione Internet attiva. 	N.I.

Codice	Requisito	Descrizione	Stato
TA1.5	L'utente deve poter verificare che venga segnalato un errore qualora segua un percorso differente da quello calcolato dall'applicazione.	All'utente è chiesto di: <ul style="list-style-type: none">• avviare la navigazione;• seguire un percorso differente da quello proposto dall'applicazione;• verificare che venga segnalato un errore che spieghi che il percorso che si sta seguendo non è quello previsto.	N.I.
TA1.6	L'utente deve poter verificare che venga segnalato un errore nel caso in cui voglia avviare la navigazione all'interno di un'area dove non è presente il segnale di alcun beacon.	All'utente è chiesto di: <ul style="list-style-type: none">• posizionarsi in un'area dove non è presente il segnale di alcun beacon;• avviare la navigazione;• verificare che venga segnalato un errore che spieghi che in quell'area non è stato rilevato il segnale di alcun beacon.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.7	L'utente deve poter verificare che venga segnalato un errore nel caso in cui voglia avviare la navigazione con la connessione Internet del proprio dispositivo non attiva.	All'utente è chiesto di: <ul style="list-style-type: none">• disattivare la connessione internet;• avviare la navigazione;• verificare che venga segnalato un errore che spieghi che il dispositivo non ha la connessione Internet attiva.	N.I.
TA1.8	L'utente deve poter verificare che venga segnalato un errore nel caso in cui voglia avviare la navigazione e la mappa installata nel proprio dispositivo differisce dall'ultima versione online della mappa.	All'utente è chiesto di: <ul style="list-style-type: none">• non aggiornare una mappa che richieda un aggiornamento;• avviare la navigazione;• verificare che venga segnalato un errore che spieghi che la mappa presente nel dispositivo non è l'ultima versione della mappa per quell'edificio.	N.I.

Codice	Requisito	Descrizione	Stato
TA1.9	L'utente deve poter verificare che venga segnalato un errore nel caso in cui si rilevi un beacon all'interno di un edificio mappato e non sia installata la mappa per quell'edificio.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • entrare in un edificio di cui non dispone della mappa; • avviare l'applicazione; • verificare che venga segnalato un errore che spieghi che non è presente nel dispositivo una mappa per quell'edificio. 	N.I.
TA2	L'utente deve poter verificare che sia possibile accedere alle informazioni dell'edificio in cui ci si trova.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • scegliere di accedere alle informazioni generali sull'edificio in cui ci si trova; • scegliere di accedere alla lista completa di tutti i POI presenti nell'edificio in cui si trova; • scegliere di accedere alla lista dei POI associati ai beacon rilevati alla posizione dell'utente. 	N.I.
TA2.1	L'utente deve poter verificare che sia possibile accedere al nome dell'edificio.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia presente un nome per l'edificio. 	N.I.

Codice	Requisito	Descrizione	Stato
TA2.2	L'utente deve poter verificare che sia possibile accedere alla descrizione dell'edificio.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia presente una descrizione per l'edificio.	N.I.
TA2.3	L'utente deve verificare che sia possibile accedere all'indirizzo dell'edificio.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia presente l'indirizzo per l'edificio.	N.I.
TA2.4	L'utente deve verificare che sia possibile accedere agli orari dell'edificio.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che siano presenti gli orari di apertura dell'edificio.	N.I.
TA2.5	L'utente deve poter verificare che sia possibile accedere alla lista completa di tutti i POI presenti nell'edificio in cui si trova.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che venga visualizzata la lista completa di tutti i POI presenti nell'edificio.	N.I.

Codice	Requisito	Descrizione	Stato
TA2.6	L'utente deve poter verificare che sia possibile accedere alla lista dei POI associati ai beacon rilevati alla posizione dell'utente.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che venga visualizzata la lista dei POI associati ai beacon rilevati alla posizione dell'utente; • verificare che sia possibile accedere alle informazioni riguardanti uno specifico POI nella lista. 	N.I.
TA2.6.1	L'utente deve poter verificare che sia possibile accedere alle informazioni riguardanti uno specifico POI.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia possibile accedere all'identificativo del POI; • verificare che sia possibile accedere alla descrizione del POI. 	N.I.
TA2.6.1.1	L'utente deve poter verificare che sia possibile accedere all'identificativo di uno specifico POI.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia presente un identificativo per il POI. 	N.I.
TA2.6.1.2	L'utente deve poter verificare che sia possibile accedere alla descrizione di uno specifico POI.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia presente una descrizione per il POI. 	N.I.

Codice	Requisito	Descrizione	Stato
TA2.7	L'utente deve verificare che venga visualizzato un errore nel caso in cui acceda alle informazioni di un edificio con connessione internet non attiva.	All'utente è chiesto di: <ul style="list-style-type: none">• disattivare la connessione internet;• accedere alle informazioni di un edificio;• verificare che venga segnalato un errore che spieghi che il dispositivo non ha la connessione Internet attiva.	N.I.
TA2.8	L'utente deve verificare che venga visualizzato un errore nel caso in cui acceda alle informazioni dell'edificio e la versione della mappa non coincida con l'ultima versione disponibile.	All'utente è chiesto di: <ul style="list-style-type: none">• non aggiornare una mappa che richieda un aggiornamento;• accedere alle informazioni di un edificio;• verificare che venga segnalato un errore che spieghi che il dispositivo non è presenta l'ultima versione di mappa disponibile.	N.I.
TA3	L'utente deve poter verificare che sia possibile gestire gli aspetti relativi all'applicazione.	All'utente è chiesto di: <ul style="list-style-type: none">• gestire le mappe dell'applicazione;• gestire le preferenze di navigazione.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1	L'utente deve poter verificare che sia possibile gestire le mappe dall'applicazione.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • scegliere di gestire le mappe installate sul proprio dispositivo; • scegliere di gestire le mappe non presenti sul proprio dispositivo. 	N.I.
TA3.1.1	l'utente deve poter verificare che sia possibile gestire le mappe presenti sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • scegliere di accedere alle mappe installate; • scegliere di aggiornare una mappa installata; • scegliere di rimuovere una mappa installata; • scegliere di accedere alle informazioni riguardanti una mappa. 	N.I.
TA3.1.1.1	L'utente deve poter verificare che sia possibile accedere alle mappe installate sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • accedere alle mappe installate; • se l'utente non ha installato alcuna mappa in precedenza verificare che la sezione sia vuota, in caso contrario verificare che la sezione contenga le mappe installate in precedenza. 	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.1.2	L'utente deve poter verificare che sia possibile aggiornare una mappa (che richieda un aggiornamento) presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">● scegliere una mappa (che richieda un aggiornamento) presente sul proprio dispositivo;● aggiornare tale mappa;● verificare che sia possibile avviare la navigazione all'interno dell'edificio di cui è stata aggiornata la mappa.	N.I.
TA3.1.1.3	L'utente deve poter verificare che sia possibile rimuovere una mappa dal proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">● scegliere una mappa tra quelle presenti sul proprio dispositivo;● rimuovere la mappa scelta;● verificare che la mappa rimossa non sia più presente sul proprio dispositivo.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.1.4	L'utente deve poter verificare che sia possibile accedere alle informazioni riguardanti una mappa presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">● scegliere una mappa presente sul proprio dispositivo;● scegliere di accedere al nome di una mappa;● scegliere di accedere alla foto associata ad una mappa;● scegliere di accedere all'indirizzo dell'edificio;● scegliere di accedere alla descrizione dell'edificio;● scegliere di accedere alla dimensione in megabyte della mappa;● scegliere di accedere alla versione della mappa.	N.I.
TA3.1.1.4.1	L'utente deve poter verificare che sia possibile accedere al nome di una mappa presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">● verificare che sia possibile accedere al nome di una mappa.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.1.4.2	L'utente deve poter verificare che sia possibile accedere all'indirizzo dell'edificio dalla mappa presente sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia possibile accedere all'indirizzo dell'edificio (dalla mappa). 	N.I.
TA3.1.1.4.3	L'utente deve poter verificare che sia possibile accedere alla descrizione dell'edificio dalla mappa presente sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia possibile accedere alla descrizione dell'edificio (dalla mappa). 	N.I.
TA3.1.1.4.4	L'utente deve poter verificare che sia possibile accedere alla dimensione in megabyte della mappa di un edificio presente sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia possibile accedere alla dimensione in megabyte della mappa di un edificio. 	N.I.
TA3.1.1.4.5	L'utente deve poter verificare che sia possibile accedere alla versione della mappa di un edificio presente sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • verificare che sia possibile accedere alla versione della mappa di un edificio. 	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.2	L'utente deve poter verificare che sia possibile gestire le mappe non presenti sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • scegliere di ricercare una mappa non presente sul proprio dispositivo; • scegliere di installare una mappa non presente sul proprio dispositivo; • scegliere di accedere alle informazioni riguardanti una mappa non presente sul proprio dispositivo. 	N.I.
TA3.1.2.1	L'utente deve poter verificare che sia possibile ricercare per nome (dell'edificio) una mappa non presente sul proprio dispositivo.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • inserire il nome dell'edificio di cui cerca la mappa; • scegliere la mappa tra quelle proposte come risultati della ricerca. 	N.I.
TA3.1.2.1.1	L'utente deve poter verificare che venga segnalato un messaggio di errore nel caso in cui l'utente voglia scaricare una mappa non prevista.	<p>All'utente è chiesto di:</p> <ul style="list-style-type: none"> • inserire il nome di una mappa non prevista dal sistema; • verificare che venga visualizzato un messaggio di errore che spieghi che tale mappa non è prevista. 	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.2.1.2	L'utente deve poter verificare che sia possibile inserire il possibile nome di una mappa.	All'utente è chiesto di: <ul style="list-style-type: none">● inserire il possibile nome di una mappa;● verificare che il nome voluto sia stato inserito.	N.I.
TA3.1.2.2	L'utente deve poter verificare che sia possibile installare una nuova mappa.	All'utente è chiesto di: <ul style="list-style-type: none">● ricercare una mappa;● scegliere una mappa tra quelle proposte nei risultati della ricerca;● eseguire il download della mappa;● verificare che la mappa sia presente tra quelle disponibili nel dispositivo;● verificare che sia possibile avviare la navigazione all'interno dell'edificio di cui è stato eseguito il download della mappa.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.2.3	L'utente deve poter verificare che sia possibile accedere alle informazioni riguardanti una mappa non ancora scaricata.	All'utente è chiesto di: <ul style="list-style-type: none">• effettuare la ricerca di una mappa;• scegliere una mappa tra i risultati della ricerca;• scegliere di accedere al nome dell'edificio;• scegliere di accedere alle foto riguardanti l'edificio;• scegliere di accedere all'indirizzo dell'edificio;• scegliere di accedere alla descrizione dell'edificio;• scegliere di accedere alla dimensione in megabyte della mappa;• scegliere di accedere alla versione della mappa.	N.I.
TA3.1.2.3.1	L'utente deve poter verificare che sia possibile accedere al nome dell'edificio dalla mappa non presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia possibile accedere al nome dell'edificio.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.1.2.3.2	L'utente deve poter verificare che sia possibile accedere all'indirizzo dell'edificio dalla mappa non presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia possibile accedere all'indirizzo dell'edificio.	N.I.
TA3.1.2.3.3	L'utente deve poter verificare che sia possibile accedere alla descrizione dell'edificio dalla mappa non presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia possibile accedere alla descrizione dell'edificio.	N.I.
TA3.1.2.3.4	L'utente deve poter verificare che sia possibile accedere alla dimensione in megabyte della mappa di un edificio non presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia possibile accedere alla dimensione in megabyte della mappa di un edificio.	N.I.
TA3.1.2.3.5	L'utente deve poter verificare che sia possibile accedere alla versione della mappa di un edificio non presente sul proprio dispositivo.	All'utente è chiesto di: <ul style="list-style-type: none">• verificare che sia possibile accedere alla versione della mappa di un edificio.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.2	L'utente deve poter verificare che sia possibile gestire le preferenze di navigazione.	All'utente è chiesto di: <ul style="list-style-type: none">• modificare le preferenze riguardanti la modalità di fruizione delle indicazioni;• modificare le preferenze riguardanti il percorso.	N.I.
TA3.2.1	L'utente deve poter verificare che sia possibile gestire le preferenze riguardanti la modalità di fruizione delle indicazioni.	All'utente è chiesto di: <ul style="list-style-type: none">• modificare le impostazioni riguardanti le indicazioni vocali;• modificare le impostazioni riguardanti le indicazioni sonore.	N.I.
TA3.2.1.1	L'utente deve poter verificare che sia possibile attivare le indicazioni vocali, se queste sono disattivate.	All'utente è chiesto di: <ul style="list-style-type: none">• attivare le indicazioni vocali;• verificare che all'avvio della navigazione vengano fornite le indicazioni vocali per raggiungere la destinazione scelta.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.2.1.2	L'utente deve poter verificare che sia possibile disattivare le indicazioni vocali, se queste sono attivate.	All'utente è chiesto di: <ul style="list-style-type: none">• disattivare le indicazioni vocali;• verificare che all'avvio della navigazione non vengano fornite le indicazioni vocali per raggiungere la destinazione scelta.	N.I.
TA3.2.1.3	L'utente deve poter verificare che sia possibile attivare le indicazioni sonore, se queste sono disattivate.	All'utente è chiesto di: <ul style="list-style-type: none">• attivare le indicazioni sonore;• verificare che all'avvio della navigazione vengano fornite le indicazioni sonore per raggiungere la destinazione scelta.	N.I.
TA3.2.1.4	L'utente deve poter verificare che sia possibile disattivare le indicazioni sonore, se queste sono attivate.	All'utente è chiesto di: <ul style="list-style-type: none">• disattivare le indicazioni sonore;• verificare che all'avvio della navigazione non vengano fornite le indicazioni sonore per raggiungere la destinazione scelta.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.2.2	L'utente deve poter verificare che sia possibile gestire le preferenze riguardanti il percorso da seguire.	All'utente viene chiesto di: <ul style="list-style-type: none">• modificare le impostazioni riguardanti il percorso più accessibile;• modificare le impostazioni riguardanti il percorso con meno ascensori.	N.I.
TA3.2.2.1	L'utente deve poter verificare che sia possibile scegliere di seguire il percorso più accessibile per arrivare alla destinazione desiderata.	All'utente viene chiesto di: <ul style="list-style-type: none">• attivare l'impostazione riguardante il percorso più accessibile;• verificare che all'avvio della navigazione l'applicazione fornisca un percorso che prediliga gli ascensori rispetto altre soluzioni per raggiungere la destinazione scelta.	N.I.

Codice	Requisito	Descrizione	Stato
TA3.2.2.2	L'utente deve poter verificare che sia possibile scegliere di seguire il percorso con il minor numero di ascensori possibile.	All'utente viene chiesto di: <ul style="list-style-type: none">• attivare l'impostazione riguardante il percorso con il minor numero di ascensori possibile;• verificare che all'avvio della navigazione l'applicazione fornisca un percorso che prediliga soluzioni alternative rispetto agli ascensori per raggiungere la destinazione scelta.	N.I.
TA3.2.2.3	L'utente deve poter verificare che sia possibile scegliere di seguire il percorso più veloce in assoluto.	All'utente viene chiesto di: <ul style="list-style-type: none">• attivare l'impostazione riguardante il percorso che è ritenuto più veloce;• verificare che all'avvio della navigazione l'applicazione fornisca un percorso che prediliga soluzioni alternative rispetto al percorso più veloce per raggiungere la destinazione scelta.	N.I.

Codice	Requisito	Descrizione	Stato
TA4	L'utente deve poter verificare che sia possibile accedere alla guida.	All'utente viene chiesto di: <ul style="list-style-type: none">• verificare che sia possibile accedere alla guida;• verificare che la guida spieghi il funzionamento dell'applicazione.	N.I.
TA5	L'utente non sviluppatore deve poter verificare che sia possibile attivare le funzionalità sviluppatore.	All'utente non sviluppatore viene chiesto di: <ul style="list-style-type: none">• inserire un codice sviluppatore valido;• confermare il codice inserito;• verificare che siano state attivate le funzionalità sviluppatore.	N.I.

Codice	Requisito	Descrizione	Stato
TA5.1	L'utente non sviluppatore deve poter verificare che venga segnalato un errore nel caso in cui venga inserito un codice sviluppatore non valido.	All'utente non sviluppatore viene chiesto di: <ul style="list-style-type: none">● inserire un codice sviluppatore non valido;● confermare il codice inserito;● verificare che venga visualizzato un errore che spieghi che il codice inserito non è valido;● verificare che non siano state attivate le funzionalità di sviluppatore.	N.I.
TA5.2	L'utente non sviluppatore deve poter verificare che sia possibile inserire un codice sviluppatore.	All'utente non sviluppatore viene chiesto di: <ul style="list-style-type: none">● inserire un codice sviluppatore;● verificare che il codice voluto sia stato inserito.	N.I.

Codice	Requisito	Descrizione	Stato
TA5.3	L'utente non sviluppatore deve poter verificare che sia possibile confermare il codice inserito.	<p>All'utente non sviluppatore viene chiesto di:</p> <ul style="list-style-type: none">● inserire un codice sviluppatore;● confermare il codice inserito;● verificare che, se il codice inserito è valido, sono ora attive le funzionalità sviluppatore, altrimenti se non è valido viene segnalato un errore.	N.I.

Codice	Requisito	Descrizione	Stato
TA6	Lo sviluppatore deve verificare che sia possibile accedere alle informazioni riguardanti i beacon rilevati.	<p>Allo sviluppatore viene chiesto di:</p> <ul style="list-style-type: none">● accedere all'UUID di un beacon rilevato;● accedere al Major di un beacon rilevato;● accedere al Minor di un beacon rilevato;● accedere al livello di potenza del segnale di un beacon rilevato;● accedere al livello di batteria di un beacon rilevato;● accedere alla distanza approssimativa dal dispositivo utilizzato al beacon rilevato;● accedere al formato di un beacon rilevato;● accedere all'area coperta da un beacon rilevato.	N.I.

Codice	Requisito	Descrizione	Stato
TA6.1	Lo sviluppatore deve verificare che sia possibile accedere all'UUID di un beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● accedere all'UUID di un beacon rilevato;● verificare che l'UUID rilevato corrisponda al valore corretto.	N.I.
TA6.2	Lo sviluppatore deve verificare che sia possibile accedere al Major di un beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● accedere al Major di un beacon rilevato;● verificare che il Major rilevato corrisponda al valore corretto.	N.I.
TA6.3	Lo sviluppatore deve verificare che sia possibile accedere al Minor di un beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● accedere al Minor di un beacon rilevato;● verificare che il Minor rilevato corrisponda al valore corretto.	N.I.
TA6.4	Lo sviluppatore deve verificare che sia possibile accedere al formato di un beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● accedere al formato di un beacon rilevato;● verificare che il formato rilevato corrisponda al valore corretto.	N.I.

Codice	Requisito	Descrizione	Stato
TA6.5	Lo sviluppatore deve verificare che sia possibile accedere al livello di potenza del segnale di un beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">• accedere al livello di potenza del segnale di un beacon rilevato.	N.I.
TA6.6	Lo sviluppatore deve verificare che sia possibile accedere al livello di batteria di un beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">• accedere al livello di batteria di un beacon rilevato.	N.I.
TA6.7	Lo sviluppatore deve verificare che sia possibile accedere alla distanza approssimativa dal dispositivo utilizzato al beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">• accedere alla distanza approssimativa dal dispositivo utilizzato al beacon rilevato.	N.I.
TA6.8	Lo sviluppatore deve verificare che sia possibile accedere all'area coperta da beacon rilevato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">• accedere all'area coperta da beacon rilevato.	N.I.

Codice	Requisito	Descrizione	Stato
TA7	Lo sviluppatore deve poter verificare che sia possibile gestire i log.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● Avviare un nuovo log;● Interrompere un log precedentemente avviato;● Accedere ad un log salvato in precedenza;● rimuovere un log salvato in precedenza;● salvare un log appena interrotto.	N.I.
TA7.1	Lo sviluppatore deve poter verificare che sia possibile avviare un nuovo log.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● avviare un nuovo log;● verificare che il log sia stato avviato.	N.I.
TA7.2	Lo sviluppatore deve poter verificare che sia possibile interrompere precedentemente avviato.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● scegliere di interrompere un log precedentemente avviato;● verificare che il log non sia più avviato.	N.I.

Codice	Requisito	Descrizione	Stato
TA7.3	Lo sviluppatore deve poter verificare che sia possibile accedere ad un log salvato in precedenza.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● accedere ad un log salvato in precedenza;● verificare che riesca a accedere al contenuto del log scelto.	N.I.
TA7.4	Lo sviluppatore deve poter verificare che sia possibile rimuovere un log salvato in precedenza.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● rimuovere un log salvato in precedenza;● verificare che il log rimosso non sia più presente nella lista dei log salvati.	N.I.
TA7.5	Lo sviluppatore deve poter verificare che sia possibile salvare un log appena interrotto.	Allo sviluppatore viene chiesto di: <ul style="list-style-type: none">● avviare un nuovo log;● interrompere il log precedentemente avviato;● salvare il log appena interrotto;● verificare che sia possibile accedere al log appena salvato.	N.I.

Tabella 2: Tabella test di accettazione

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_g per verificare il corretto funzionamento del software_g ed evidenziare eventuali bug_g o mancanze a livello di performance e precisione.

Di seguito vengono riportati i test di sistema definiti dal gruppo *Leaf*.

Test	Descrizione	Requisito	Stato
TS1	Viene verificato che il sistema calcoli un percorso per navigare da un POI A ad un POI B.	RObbF7.3	N.I.
TS1.1	Viene verificato che il sistema calcoli un percorso per navigare da un POI A ad un POI B secondo le preferenze dell'utente.	RDesF7.3.1	N.I.
TS1.1.1	Viene verificato che il sistema calcoli un percorso per navigare da un POI A ad un POI B scegliendo il percorso con meno barriere architettoniche.	RDesF7.3.1.1	N.I.
TS1.1.2	Viene verificato che il sistema calcoli un percorso per navigare da un POI A ad un POI B scegliendo il percorso con meno ascensori.	RDesF7.3.1.2	N.I.

Test	Descrizione	Requisito	Stato
TS1.1.3	Viene verificato che il sistema calcoli un percorso per navigare da un POI A ad un POI B scegliendo il percorso più veloce.	RDesF7.3.1.3	N.I.
TS1.2	Viene verificato che il sistema fornisca le indicazioni per raggiungere il prossimo POI.	ROpzF7.4.2.4	N.I.
TS1.3	Viene verificato che il sistema fornisce una lista contenente le indicazioni utili per raggiungere la destinazione scelta percorrendo tutti i POI che compongono il percorso previsto.	RDesF7.4.2.2	N.I.
TS1.4	Viene verificato che il sistema avvisi l'utente qualora rilevi un beacon differente da quelli previsti dal percorso calcolato.	RDesF7.4.2.3	N.I.
TS1.5	Viene verificato che il sistema avvisi l'utente qualora si trovi in un'area in cui non viene rilevato alcun beacon.	ROpzF7.4.2.6	N.I.
TS1.6	Viene verificato che il sistema fornisca delle informazioni testuali estese.	ROpzF7.4.3.2	N.I.
TS1.7	Viene verificato che il sistema fornisca le fotografie del prossimo POI da raggiungere.	RDesF7.4.3.1	N.I.

Test	Descrizione	Requisito	Stato
TS1.8	Viene verificato che il sistema fornisca la lista di tutte le prossime indicazioni da seguire per raggiungere la destinazione scelta.	ROpzF7.4.3.3	N.I.
TS1.9	Viene verificato che il sistema permetta di interrompere la navigazione in corso.	RObbF7.5	N.I.
TS1.9.1	Viene verificato che il sistema richieda l'attivazione della geolocalizzazione.	RObbF7.4.1.1	N.I.
TS1.9.2	Viene verificato che il sistema richieda l'attivazione del Bluetooth.	RObbF7.4.1.2	N.I.
TS1.9.3	Viene verificato che il sistema richieda l'attivazione del GPS se il dispositivo ha una versione del sistema operativo uguale o superiore a 6.0.	RObbF7.4.1.3	N.I.
TS1.10	Viene verificato che il sistema avverta l'utente qualora volesse avviare la navigazione in mancanza di una connessione internet attiva.	RObbF7.6	N.I.
TS1.11	Viene verificato che il sistema avverta l'utente qualora volesse avviare la navigazione e la mappa installata sul suo dispositivo differisce dall'ultima versione disponibile per quell'edificio.	RObbF7.7	N.I.

Test	Descrizione	Requisito	Stato
TS1.12	Viene verificato che il sistema avverta l'utente qualora rilevasse un beacon all'interno di un edificio e la mappa dell'edificio non fosse installata nel dispositivo.	RObbF7.8	N.I.
TS1.13	Viene verificato che il sistema fornisca la possibilità di ricercare una destinazione per nome.	RDesF7.1.1	N.I.
TS1.13.1	Viene verificato che il sistema fornisca la possibilità di inserire il nome di una destinazione.	RDesF7.1.1.1	N.I.
TS1.14	Viene verificato che il sistema fornisca la possibilità di ricercare una destinazione per categoria.	RObbF7.1.2	N.I.
TS1.14.1	Viene verificato che il sistema permetta di accedere ad una categoria tra quelle disponibili per il dato edificio, accedendo ai POI in essa contenuti.	RObbF7.1.2.1	N.I.
TS1.15	Viene verificato che il sistema permetta di selezionare il risultato di una ricerca.	RObbF7.1.3	N.I.
TS1.16	Viene verificato che il sistema permetta di confermare la scelta di una destinazione.	RObbF7.2	N.I.

Test	Descrizione	Requisito	Stato
TS1.17	Viene verificato che il sistema avverta l'utente qualora volesse accedere alla foto del prossimo POI e la connessione Internet non fosse attiva sul proprio dispositivo.	RDesF7.4.3.4	N.I.
TS2	Viene verificato che il sistema richieda l'attivazione dei sensori.	RObbF7.4.1	N.I.
TS3	Viene verificato che il sistema interagisca con i beacon.	RObbF8	N.I.
TS3.1	Viene verificato che il sistema rilevi gli identificativi (UUID, Major, Minor) di un beacon rilevato dall'applicazione.	RObbF8.1	N.I.
TS3.1.1	Viene verificato che, rilevato l'identificativo di un beacon, il sistema riesca a reperire informazioni riguardanti il POI a cui è associato quel beacon.	RObbF8.1.1	N.I.
TS3.1.2	Viene verificato che, rilevato l'identificativo di un beacon, il sistema riesca a reperire informazioni riguardanti i POI circostanti quel beacon.	RObbF8.1.2	N.I.
TS3.2	Viene verificato che il sistema rilevi il livello di potenza del segnale di un beacon rilevato.	RObbF8.2	N.I.

Test	Descrizione	Requisito	Stato
TS3.3	Viene verificato che il sistema rilevi il livello di batteria di un beacon rilevato.	RObbF8.3	N.I.
TS3.4	Viene verificato che il sistema rilevi la distanza approssimativa di un beacon rilevato dal dispositivo utilizzato.	RObbF8.4	N.I.
TS3.5	Viene verificato che il sistema rilevi il formato di un beacon rilevato.	RObbF8.5	N.I.
TS3.6	Viene verificato che il sistema rilevi l'area coperta dal segnale di un beacon rilevato.	RObbF8.6	N.I.
TS4	Viene verificato che il sistema permetta di recuperare una mappa collegandosi ad un server.	RDesF10.2.3	N.I.
TS5	Viene verificato che il sistema permetta di accedere al nome dell'edificio in cui si trova l'utente.	RObbF9.5	N.I.
TS6	Viene verificato che il sistema permetta di accedere alla descrizione dell'edificio in cui si trova l'utente.	RObbF9.6	N.I.
TS7	Viene verificato che il sistema permetta di accedere all'orario dell'edificio in cui si trova l'utente.	RObbF9.3	N.I.
TS8	Viene verificato che il sistema permetta di accedere all'indirizzo dell'edificio in cui si trova l'utente.	RObbF9.4	N.I.

Test	Descrizione	Requisito	Stato
TS9	Viene verificato che il sistema permetta di accedere alla lista di POI di un edificio.	RObbF9.1	N.I.
TS10	Viene verificato che il sistema permetta di accedere alle informazioni relative ad uno specifico POI.	ROpzF9.2.1	N.I.
TS10.1	Viene verificato che il sistema permetta di accedere al nome di un POI.	RObbF9.2.3	N.I.
TS10.2	Viene verificato che il sistema permetta di accedere alla descrizione di un POI.	RObbF9.2.4	N.I.
TS11	Viene verificato che il sistema permetta di accedere ad un elenco dei POI appartenenti all'edificio in cui si trova l'utente e rilevati alla posizione dell'utente.	ROpzF9.2.2	N.I.
TS12	Viene verificato che il sistema avverta l'utente qualora volesse accedere alle informazioni dell'edificio in cui si trova e la connessione Internet non fosse attiva sul proprio dispositivo.	RObbF9.9	N.I.
TS13	Viene verificato che il sistema avverta l'utente qualora volesse accedere alle informazioni dell'edificio in cui si trova e la versione della mappa presente sul dispositivo non coincidesse con l'ultima versione della mappa disponibile.	RObbF9.10	N.I.

Test	Descrizione	Requisito	Stato
TS14	Viene verificato che il sistema permetta di impostare le preferenze di navigazione.	ROpzF10.1	N.I.
TS14.1	Viene verificato che il sistema permetta di fornire le indicazioni in forma testuale.	RObbF10.1.2.1	N.I.
TS14.2	Viene verificato che il sistema permetta di attivare le indicazioni sonore.	RDesF10.1.2.3	N.I.
TS14.3	Viene verificato che il sistema permetta di attivare le indicazioni vocali.	RDesF10.1.2.2	N.I.
TS14.4	Viene verificato che il sistema permetta di disattivare le indicazioni sonore.	RDesF10.1.2.5	N.I.
TS14.5	Viene verificato che il sistema permetta di disattivare le indicazioni vocali.	RDesF10.1.2.4	N.I.
TS14.6	Viene verificato che il sistema permetta di scegliere il percorso più accessibile.	ROpzF10.1.1.1	N.I.
TS14.7	Viene verificato che il sistema permetta di scegliere il percorso con il minor numero di ascensori.	ROpzF10.1.1.2	N.I.
TS14.8	Viene verificato che il sistema permetta di scegliere il percorso più veloce.	RObbF10.1.1.3	N.I.
TS15	Viene verificato che il sistema permetta la gestione delle mappe.	RDesF10.2	N.I.

Test	Descrizione	Requisito	Stato
TS15.1	Viene verificato che il sistema permetta di accedere alle mappe installate nel proprio dispositivo.	RDesF10.2.1.1	N.I.
TS15.2	Viene verificato che il sistema permetta di installare una mappa disponibile online non precedentemente installata.	RDesF10.2.2.2	N.I.
TS15.3	Viene verificato che il sistema permetta di ricercare una mappa.	RDesF10.2.2.1	N.I.
TS15.4	Viene verificato che il sistema permetta di rimuovere una mappa.	RDesF10.2.1.3	N.I.
TS15.5	Viene verificato che il sistema permetta di aggiornare una mappa.	RDesF10.2.1.2	N.I.
TS15.6	Viene verificato che il sistema permetta di accedere al nome di una mappa presente sul dispositivo.	RDesF10.2.1.4.1	N.I.
TS15.7	Viene verificato che il sistema permetta di accedere all'indirizzo dell'edificio a cui si riferisce una mappa presente sul proprio dispositivo.	RDesF10.2.1.4.2	N.I.
TS15.8	Viene verificato che il sistema permetta di accedere alla descrizione dell'edificio a cui si riferisce una mappa presente sul proprio dispositivo.	RDesF10.2.1.4.3	N.I.

Test	Descrizione	Requisito	Stato
TS15.9	Viene verificato che il sistema permetta di accedere alla dimensione in megabyte di una mappa presente sul proprio dispositivo.	RDesF10.2.1.4.4	N.I.
TS15.10	Viene verificato che il sistema permetta di accedere alla versione di una mappa presente sul proprio dispositivo.	RDesF10.2.1.4.5	N.I.
TS15.11	Viene verificato che il sistema permetta di accedere al nome di una mappa non presente sul dispositivo.	RDesF10.2.2.3.1	N.I.
TS15.12	Viene verificato che il sistema permetta di accedere all'indirizzo dell'edificio a cui si riferisce una mappa non presente sul proprio dispositivo.	RDesF10.2.2.3.2	N.I.
TS15.13	Viene verificato che il sistema permetta di accedere alla descrizione dell'edificio a cui si riferisce una mappa non presente sul proprio dispositivo.	RDesF10.2.2.3.3	N.I.
TS15.14	Viene verificato che il sistema permetta di accedere alla dimensione in megabyte di una mappa non presente sul proprio dispositivo.	RDesF10.2.2.3.4	N.I.

Test	Descrizione	Requisito	Stato
TS15.15	Viene verificato che il sistema permetta di accedere alla versione di una mappa non presente sul proprio dispositivo.	RDesF10.2.2.3.5	N.I.
TS15.16	Viene verificato che il sistema segnali all'utente qualora la ricerca per nome non abbia trovato corrispondenza tra le mappe disponibili online..	RDesF10.2.2.4	N.I.
TS16	Viene verificato che il sistema permetta di inserire il codice sviluppatore.	RObbF10.3.1	N.I.
TS16.1	Viene verificato che il sistema permetta di confermare il codice sviluppatore.	RObbF10.3.2	N.I.
TS17	Viene verificato che il sistema metta a disposizione una sezione per la guida.	ROpzF11	N.I.
TS18	Viene verificato che il sistema permetta di accedere alle informazioni di un beacon rilevato.	RObbF12	N.I.
TS18.1	Viene verificato che il sistema permetta di accedere al UUID di un beacon rilevato.	RObbF12.1	N.I.
TS18.2	Viene verificato che il sistema permetta di accedere al Major di un beacon rilevato.	RObbF12.8	N.I.

Test	Descrizione	Requisito	Stato
TS18.3	Viene verificato che il sistema permetta di accedere al Minor di un beacon rilevato.	RObbF12.9	N.I.
TS18.4	Viene verificato che il sistema permetta di accedere al livello di potenza del segnale di un beacon rilevato.	RObbF12.2	N.I.
TS18.5	Viene verificato che il sistema permetta di accedere al livello di batteria di un beacon rilevato.	ROpzF12.4	N.I.
TS18.6	Viene verificato che il sistema permetta di accedere alla distanza approssimativa di un beacon rilevato dal dispositivo utilizzato.	RObbF12.5	N.I.
TS18.7	Viene verificato che il sistema permetta di accedere al formato di un beacon rilevato.	RObbF12.6	N.I.
TS18.8	Viene verificato che il sistema permetta di accedere all'area coperta da un beacon rilevato.	RObbF12.7	N.I.
TS18.9	Viene verificato che il sistema permetta di gestire un log.	RObbF12.3	N.I.
TS18.9.1	Viene verificato che il sistema permetta di avviare un log.	RDesF12.3.2	N.I.
TS18.9.2	Viene verificato che il sistema permetta di interrompere un log.	RDesF12.3.1	N.I.

Test	Descrizione	Requisito	Stato
TS18.9.3	Viene verificato che il sistema permetta di salvare un log.	RDesF12.3.3	N.I.
TS18.9.4	Viene verificato che il sistema permetta di rimuovere un log.	RDesF12.3.5	N.I.
TS18.9.5	Viene verificato che il sistema permetta di accedere ad un log salvato.	RDesF12.3.4	N.I.
TS19	Viene verificato che il sistema avverta l'utente qualora venga inserita una destinazione non prevista dal sistema.	RObbF7.1.4	N.I.
TS20	Viene verificato che il sistema avverta l'utente qualora il codice inserito per sbloccare le funzionalità sviluppatore non sia corretto.	RObbF10.3.3	N.I.
TS21	Viene verificato che il sistema fornisca la possibilità di inserire il possibile nome di una mappa.	RDesF10.2.2.1.1	N.I.

Tabella 3: Tabella di tracciamento test di sistema / requisiti

D.3 Test di integrazione

I test di integrazione servono per verificare il corretto funzionamento di più moduli assemblati insieme. Per una descrizione completa della sintassi utilizzata nella descrizione di tali test si consulti il documento *Norme di progetto v4.00*.

D.3.1 Test-componenti

Test	Descrizione	Componenti	Stato
TI1	Test di integrazione finale tra tutte le componenti per verificare il corretto comportamento del sistema nel suo complesso		N.I.
TI2	Viene verificato che il sistema gestisca correttamente le componenti relative al package model. In particolare che la navigazione si comporti secondo le esigenze dell'utente, utilizzi correttamente i sensori, si interfacci correttamente con la tecnologia Beacon e vengano recuperate le informazioni dal database	model	N.I.
TI3	Viene verificato che il sistema gestisca correttamente le componenti relative al package beacon. In particolare che si interfacci correttamente con la libreria Alt-Beacon e con la tecnologia Beacon	model::beacon	N.I.
TI4	Viene verificato che il sistema gestisca correttamente le componenti relative al package usersetting. In particolare che gestisca correttamente le preferenze dell'utente e le renda persistenti sul dispositivo	model::usersetting	N.I.

Test	Descrizione	Componenti	Stato
TI5	Viene verificato che il sistema gestisca correttamente le componenti relative al package compass. In particolare che sia possibile avviare e fermare la bussola	model::compass	N.I.
TI6	Viene verificato che il sistema gestisca correttamente le componenti relative al package navigator. In particolare che fornisca le funzionalità di navigazione e calcoli il percorso secondo le esigenze dell'utente	model::navigator	N.I.
TI7	Viene verificato che il sistema gestisca correttamente le componenti relative al package algorithm. In particolare che, dati un grafo pesato e due nodi, calcoli un percorso dal nodo di partenza al nodo di arrivo	model::navigator::algorithm	N.I.
TI8	Viene verificato che il sistema gestisca correttamente le componenti relative al package graph. In particolare che gestisca la struttura di un grafo e associ delle informazioni ad un PointOfInterest o ad un Edge	model::navigator::graph	N.I.
TI9	Viene verificato che il sistema gestisca correttamente le componenti relative al package vertex. In particolare che rappresenti un nodo di un grafo.	model::navigator::graph::vertex	N.I.

Test	Descrizione	Componenti	Stato
TI10	Viene verificato che il sistema gestisca correttamente le componenti relative al package area. In particolare che rappresentino la struttura di un edificio come PointOfInterest e RegionOfInterest	model::navigator::graph:- area	N.I.
TI11	Viene verificato che il sistema gestisca correttamente le componenti relative al package navigationInformation. In particolare rappresenti i diversi tipi di istruzione a seconda delle preferenze dell'utente	model::navigator::graph:- navigationinformation	N.I.
TI12	Viene verificato che il sistema gestisca correttamente le componenti relative al package edge. In particolare che rappresenti i diversi tipi di archi e le informazioni ad essi associate	model::navigator::graph:- edge	N.I.
TI13	Viene verificato che il sistema gestisca correttamente le componenti relative al package dataaccess. In particolare che si interfacci in maniera corretta col database remoto e permetta di gestire le informazioni nel database locale	model::dataaccess	N.I.

Test	Descrizione	Componenti	Stato
TI14	Viene verificato che il sistema gestisca correttamente le componenti relative al package service. In particolare che faccia da tramite per l'accesso al database per il resto del model e costruisca oggetti della Business Logic a partire dagli oggetti che rappresentano le tabelle del database	model::dataaccess::service	N.I.
TI15	Viene verificato che il sistema gestisca correttamente le componenti relative al package dao. In particolare che sia possibile effettuare le operazioni CRUD sul database locale e tradurre la struttura del database in oggetti, che rappresentano le tabelle, per accedere alle informazioni	model::dataaccess::dao	N.I.
TI16	Viene verificato che il sistema gestisca correttamente le componenti relative al package view. In particolare che riesca a recuperare le informazioni dal package presenter e che le esponga correttamente all'utente	view	N.I.
TI17	Viene verificato che il sistema gestisca correttamente le componenti relative al package presenter. In particolare che gestisca correttamente l'interazione coi componenti del package model e del package view	presenter	N.I.

Test	Descrizione	Componenti	Stato
------	-------------	------------	-------

Tabella 4: Tabella test di integrazione**D.3.2 Componente - test**

Componente	Test
model	TI2
model::beacon	TI3
model::compass	TI5
model::dataaccess	TI13
model::dataaccess::dao	TI15
model::dataaccess::service	TI14
model::navigator	TI6
model::navigator::algorithm	TI7
model::navigator::graph	TI8
model::navigator::graph::area	TI10
model::navigator::graph::edge	TI12
model::navigator::graph::navigationinformation	TI11
model::navigator::graph::vertex	TI9
model::usersetting	TI4
presenter	TI17
view	TI16

Tabella 5: Tabella componente / test di integrazione

D.4 Test di unità

Il test di unità serve per accertare il corretto funzionamento delle singole unità, ovvero le classi. Per una descrizione completa della sintassi utilizzata nella descrizione di tali test si consulti il documento *Norme di progetto v4.00*.

Test	Descrizione	Metodi	Stato
TU1	Viene testato che tramite un oggetto SettingImp sia possibile salvare e recuperare le informazioni relative alle preferenze dell'utente	model::usersetting::- SettingImp::- getPathPreference() model::usersetting::- SettingImp::- getInstructionPreference() model::usersetting::- SettingImp::- setPathPreference() model::usersetting::- SettingImp::- setInstructionPreference()	N.I
TU2	Viene testato che tramite la classe DeveloperCodeManager sia possibile riconoscere un codice sviluppatore valida da uno non valido	model::usersetting::- DeveloperCodeManager::- isValid()	N.I
TU3	Viene testato che tramite un oggetto della classe SettingImp sia possibile recuperare e modificare le informazioni riguardanti un eventuale codice sviluppatore inserito. In particolare viene testato se vengono salvate le informazioni relative al fatto che un utente sia o meno sviluppatore	model::usersetting::- SettingImp::isDeveloper() model::usersetting::- SettingImp::- unlockDeveloper()	N.I

Test	Descrizione	Metodi	Stato
TU4	Viene testato che sia possibile, utilizzando un oggetto BuildingInformation, accedere alle informazioni relative ad un edificio	model::navigator::- BuildingInformation::- getName() model::navigator::- BuildingInformation::- getDescription() model::navigator::- BuildingInformation::- getOpeningHours() model::navigator::- BuildingInformation::- getAddress() model::navigator::- BuildingInformation::- toString()	N.I

Test	Descrizione	Metodi	Stato
TU5	Viene testato che utilizzando un oggetto BuildingMapImp sia possibile accedere alle informazioni dell'edificio, alla versione della mappa della mappa e al suo id all'interno del database, e alle collezioni di PointOfInterest, RegionOfInterest e EnrichedEdge che contiene	model::navigator::- BuildingMapImp::- getAddress() model::navigator::- BuildingMapImp::- getAllBuildingInformation() model::navigator::- BuildingMapImp::- getAllEdges() model::navigator::- BuildingMapImp::- getAllPOIs() model::navigator::- BuildingMapImp::- getAllROIs() model::navigator::- BuildingMapImp::- getDescription() model::navigator::- BuildingMapImp::getId() model::navigator::- BuildingMapImp::- getName() model::navigator::- BuildingMapImp::- getOpeningHours() model::navigator::- BuildingMapImp::- getVersion() model::navigator::- BuildingMapImp::- getSize()	N.I

Test	Descrizione	Metodi	Stato
TU6	Viene testato che utilizzando un oggetto BuildingMapImp sia possibile accedere alla collezione di PointOfInterest associati ad alla RegionOfInterest che contiene il beacon passato	model::navigator::- BuildingMapImp::- getAllNearbyPOIs()	N.I
TU7	Viene testato che sia possibile accedere a tutte le informazioni contenute in un oggetto ProcessedInformationImp	model::navigator::- ProcessedInformation::- getProcessedBasicInstruction() model::navigator::- ProcessedInformation::- getDetailedInstruction() model::navigator::- ProcessedInformation::- getPhotoInstruction()	N.I
TU8	Viene testato che sia possibile accedere alle informazioni contenute in un oggetto VertexImp	model::navigator::graph::- vertex::VertexImp::getId()	N.I
TU9	Viene testato che sia possibile accedere alle informazioni contenute in un oggetto BasicInformation	model::navigator::graph::- navigationinformation::- BasicInformation::- getBasicInformation()	N.I
TU10	Viene testato che sia possibile accedere alle informazioni contenute in un oggetto DetailedInformation	model::navigator::graph::- navigationinformation::- DetailedInformation::- getDetailedInformation()	N.I
TU11	Viene testato che sia possibile accedere alle informazioni contenute in un oggetto PhotoRef	model::navigator::graph::- navigationinformation::- PhotoRef::getPhotoUri()	N.I

Test	Descrizione	Metodi	Stato
TU12	Viene testato che sia possibile accedere alla collezione di oggetti PhotoRef contenuta in un oggetto PhotoInformation	model::navigator::graph::-navigationinformation::-PhotoInformation::-getPhotoInformation()	N.I
TU13	Viene testato che sia possibile accedere alle informazioni contenute in un oggetto NavigationInformationImp	model::navigator::graph::-navigationinformation::-NavigationInformation::-getBasicInformation() model::navigator::graph::-navigationinformation::-NavigationInformation::-getDetailedInformation() model::navigator::graph::-navigationinformation::-NavigationInformation::-getPhotoInformation()	N.I
TU14	Viene testato che sia possibile accedere alle informazioni relative punto di inizio, punto di fine, distanza tra i due punti, l'angolo, rispetto al nord, che c'è tra il primo e il secondo punto, collezione di PhotoRef, id di un oggetto che ha tipo statico AbsEnrichedEdge e tipo dinamico un sottotipo di AbsEnrichedEdge	model::navigator::graph::-edge::AbsEnrichedEdge::-getStarterPoint() model::navigator::graph::-edge::AbsEnrichedEdge::-getEndPoint() model::navigator::graph::-edge::AbsEnrichedEdge::-getPhotoInformation() model::navigator::graph::-edge::AbsEnrichedEdge::-getCoordinate() model::navigator::graph::-edge::AbsEnrichedEdge::-getId()	N.I
TU15	Viene testato che sia possibile impostare le preferenze relative agli archi da attraversare	model::navigator::graph::-edge::AbsEnrichedEdge::-setUserPreference()	N.I

Test	Descrizione	Metodi	Stato
TU16	Viene testato che sia possibile accedere alle informazioni di navigazione all'interno di una sottoclasse di <code>AbsEnrichedEdge</code>	<code>model::navigator::graph::edge::AbsEnrichedEdge::getNavigationInformation()</code>	N.I
TU17	Viene testato che sia possibile accedere alle informazioni di base per superare tale arco e quelle dettagliate in un oggetto <code>DefaultEdge</code>	<code>model::navigator::graph::edge::DefaultEdge::getBasicInformation()</code> <code>model::navigator::graph::edge::DefaultEdge::getDetailedInformation()</code>	N.I
TU18	Viene testato che sia possibile accedere alle informazioni di base per superare tale arco e quelle dettagliate in un oggetto <code>StairEdge</code>	<code>model::navigator::graph::edge::StairEdge::getBasicInformation()</code> <code>model::navigator::graph::edge::StairEdge::getDetailedInformation()</code>	N.I
TU19	Viene testato che sia possibile accedere alle informazioni di base per superare tale arco e quelle dettagliate in un oggetto <code>ElevatorEdge</code>	<code>model::navigator::graph::edge::ElevatorEdge::getBasicInformation()</code> <code>model::navigator::graph::edge::ElevatorEdge::getDetailedInformation()</code>	N.I
TU20	Viene testato che il peso dell'arco venga calcolato in base alle preferenze impostate tramite il metodo <code>AbsEnrichedEdge.setUserPreference</code> in un oggetto <code>StairEdge</code>	<code>model::navigator::graph::edge::StairEdge::getWeight()</code>	N.I

Test	Descrizione	Metodi	Stato
TU21	Viene testato che il peso dell'arco venga calcolato in base alle preferenze impostate tramite il metodo AbsEnrichedEdge.setUserPreference in un oggetto ElevatorEdge	model::navigator::graph::- edge::ElevatorEdge::- getWeight()	N.I
TU22	Viene testato che il peso dell'arco venga calcolato in base alle preferenze impostate tramite il metodo AbsEnrichedEdge.setUserPreference in un oggetto DefaultEdge	model::navigator::graph::- edge::DefaultEdge::- getWeight()	N.I
TU23	Viene testato che sia possibile accedere a tutte le informazioni contenute in un oggetto PointOfInterestInformation	model::navigator::- graph::area::- PointOfInterestInformation::- getName() model::navigator::- graph::area::- PointOfInterestInformation::- getDescription() model::navigator::- graph::area::- PointOfInterestInformation::- getCategory()	N.I

Test	Descrizione	Metodi	Stato
TU24	Viene testato che sia possibile accedere a tutte le informazioni riguardanti il POI e all'id del POI relativo al database in un oggetto PointOfInterestImp	model::navigator::- graph::area::- PointOfInterestImp::- getName() model::navigator::- graph::area::- PointOfInterestImp::- getDescription() model::navigator::- graph::area::- PointOfInterestImp::- getCategory() model::navigator::- graph::area::- PointOfInterestImp::- getId()	N.I
TU25	Viene testato che sia possibile settare e accedere a tutti i ROI in cui è contenuto il POI rappresentato da un oggetto PointOfInterest	model::navigator::- graph::area::- PointOfInterestImp::- getAllBelongingROIs() model::navigator::- graph::area::- PointOfInterestImp::- setBelongingROIs()	N.I
TU26	Viene testato che sia possibile accedere alle informazioni alle informazioni relative al beacon che è contenuto in una determinata ROI tramite un oggetto RegionOfInterestImp	model::navigator::- graph::area::- RegionOfInterestImp::- getUUID() model::navigator::- graph::area::- RegionOfInterestImp::- getMajor() model::navigator::- graph::area::- RegionOfInterestImp::- getMinor()	N.I

Test	Descrizione	Metodi	Stato
TU27	Viene testato che sia possibile ricavare il piano di appartenenza di un oggetto RegionOfInterestImp ricavandolo dal minor	model::navigator::- graph::area::- RegionOfInterestImp::- getFloor()	N.I
TU28	Viene testato che sia possibile verificare tramite la classe RegionOfInterestImp è possibile verificare se un beacon è contenuto o meno in una certa ROI	model::navigator::- graph::area::- RegionOfInterestImp::- contains()	N.I
TU29	Viene testato che sia possibile settare e accedere a tutti i POI contenuti nel ROI rappresentato da un oggetto RegionOfInterest	model::navigator::- graph::area::- RegionOfInterestImp::- getAllNearbyPOIs() model::navigator::- graph::area::- RegionOfInterestImp::- setNearbyPOIs()	N.I
TU30	Viene testato che sia possibile aggiungere EnrichedEdge e RegionOfInterest ad un oggetto MapGraph	model::navigator::- graph::MapGraph::- addAllRegions() model::navigator::graph::- MapGraph::addEdge() model::navigator::graph::- MapGraph::addAllEdges()	N.I
TU31	Viene testato che un oggetto MapGraph possa ritornare un grafo	model::navigator::graph::- MapGraph::getGraph()	N.I
TU32	Viene testato che sia possibile calcolare un percorso formato da una lista di Edges utilizzando un oggetto DijkstraPathFinder	model::navigator::- algorithm::- DijkstraPathFinder::- calculatePath()	N.I

Test	Descrizione	Metodi	Stato
TU33	Viene testato che sia passibile settare ad un oggetto Navigator il grafo su cui si vuole effettuare la navigazione e calcolare un percorso da un certo punto ad un altro. In particolare deve essere testato che vengano lanciata l'eccezione NoGraphSetException nel caso in cui venga richiesto di calcolare un percorso e non sia stato settato alcun grafo, mentre deve essere lanciata l'eccezione NoNavigationInformationException nel caso in cui si richieda un percorso e quest'ultimo non è ancora stato calcolato	model::navigator::- NavigatorImp::- calculatePath() model::navigator::- NavigatorImp::setGraph() model::navigator::- NavigatorImp::getPath()	N.I
TU34	Viene testato che sia possibile, settato un grafo e calcolato un percorso, ottenere tutte le istruzioni di navigazione. In particolare deve essere lanciata un'eccezione di tipo NoNavigationInformationException nel caso in cui si richiedano le informazioni riguardanti un percorso ma queste non siano disponibili poichè non è stato settato un grafo o non è ancora stato calcolato un percorso	model::navigator::- NavigatorImp::- getAllInstructions()	N.I

Test	Descrizione	Metodi	Stato
TU35	Viene testato che sia possibile, settato un grafo e calcolato un percorso, ottenere le informazioni di navigazione una di seguito all'altra. In particolare deve essere lanciata un'eccezione di tipo <code>NoNavigationInformationException</code> nel caso in cui si richiedano le informazioni riguardanti un percorso ma queste non siano disponibili poiché non è stato settato un grafo o non è ancora stato calcolato un percorso. Inoltre viene lanciata un'eccezione <code>PathException</code> nel caso in cui il beacon più potente rilevato non faccia parte del percorso previsto	<code>model::navigator::NavigatorImp::toNextRegion()</code>	N.I

Test	Descrizione	Metodi	Stato
TU36	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto MyBeacon	model::beacon::- MyBeacon::getUUID() model::beacon::- MyBeacon::getMajor() model::beacon::- MyBeacon::- getBluetoothAddress() model::beacon::- MyBeacon::getDistance() model::beacon::- MyBeacon::getTxPower() model::beacon::- MyBeacon::getRssi() model::beacon::- MyBeacon::getMinor() model::beacon::- MyBeacon::- getBatteryLevel() model::beacon::- MyBeacon::- getBeaconTypeCode()	N.I
TU37	Viene testato che sia possibile aggiungere ad un oggetto Log le informazioni di un beacon	model::beacon::Logger::- add()	N.I
TU38	Viene testato che sia possibile salvare un oggetto Log	model::beacon::Logger::- save()	N.I
TU39	Viene testato che sia possibile mettere un oggetto BeaconManagerAdapter in background mode	model::beacon::- BeaconManagerAdapter::- setBackgroundMode()	N.I
TU40	Viene testato che sia possibile modificare il periodo di scansione di un oggetto BeaconManagerAdapter	model::beacon::- BeaconManagerAdapter::- modifyScanPeriod()	N.I

Test	Descrizione	Metodi	Stato
TU41	Viene testato che sia possibile accedere alle informazioni associate ad un oggetto InformationManagerImp. In particolare nel caso in cui nessuna lista di beacon sia disponibile oppure non sia stato ancora visto un beacon e quindi non sia possibile accedere alla mappa dell'edificio deve essere lanciata un'eccezione di tipo NoBeaconSeenException	model:- InformationManagerImp:- getBuildingMap() model:- InformationManagerImp:- getNearbyPOIs() model:- InformationManagerImp:- getAllVisibleBeacons() model:- InformationManagerImp:- getDatabaseService()	N.I
TU42	Viene testato che sia possibile aggiungere e rimuovere un listener ad un oggetto NavigationManagerImp. In particolare viene anche testato che nel caso in cui sia registrato almeno un listener venga avvertito nel caso in cui venga settata una nuova lista di beacon	model:- NavigationManagerImp:- addBeaconListener() model:- NavigationManagerImp:- removeBeaconListener()	N.I
TU43	Viene testato che sia possibile registrare e salvare informazioni sui beacon rilevati utilizzando un oggetto InformationManagerImp	model:- InformationManagerImp:- startRecordingBeacons() model:- InformationManagerImp:- saveRecordedBeaconInformation()	N.I

Test	Descrizione	Metodi	Stato
TU44	Viene testato che sia possibile gestire la navigazione utilizzando un oggetto della classe NavigationManagerImp	model::- NavigationManagerImp::- startNavigation() model::- NavigationManagerImp::- getAllNavigationInstruction() model::- NavigationManagerImp::- getNextInstruction() model::- NavigationManagerImp::- stopNavigation()	N.I
TU45	Viene testato che sia possibile creare e ritornare un oggetto BuildingService.	model::dataaccess::- service::ServiceHelper::- getService()	N.I
TU46	Viene testato che sia possibile eliminare una foto dal database locale, recuperarne una o tutte quelle riguardanti un Edge.	model::dataaccess::- service::PhotoService::- deletePhoto() model::dataaccess::- service::PhotoService::- findPhoto() model::dataaccess::- service::PhotoService::- findAllPhotosOfEdge()	N.I
TU47	Viene testato che sia possibile creare un oggetto PhotoRef dato un oggetto PhotoTable.	model::dataaccess::- service::PhotoService::- fromTableToBo()	N.I
TU48	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto PhotoTable, sia possibile costruire un oggetto PhotoTable e inserirlo nel database locale.	model::dataaccess::- service::PhotoService::- convertAndInsert()	N.I

Test	Descrizione	Metodi	Stato
TU49	Viene testato che sia possibile eliminare una RegionOfInterest dal database locale, recuperarne una o tutte quelle riguardanti un edificio, dato il major del suddetto edificio.	model::dataaccess::- service::- RegionOfInterestService::- deleteRegionOfInterest() model::dataaccess::- service::- RegionOfInterestService::- findRegionOfInterest() model::dataaccess::- service::- RegionOfInterestService::- findAllRegionsWithMajor()	N.I
TU50	Viene testato che sia possibile creare un oggetto RegionOfInterestImp dato un oggetto RegionOfInterestTable.	model::dataaccess::- service::- RegionOfInterestService::- fromTableToBo()	N.I
TU51	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto RegionOfInterestTable, sia possibile costruire un oggetto RegionOfInterestTable e inserirlo nel database locale.	model::dataaccess::- service::- RegionOfInterestService::- convertAndInsert()	N.I
TU52	Viene testato che sia possibile eliminare un Edge dal database locale, recuperarne uno o tutti quelli riguardanti un edificio, dato il major del suddetto edificio.	model::dataaccess::- service::EdgeService::- deleteEdge() model::dataaccess::- service::EdgeService::- findEdge() model::dataaccess::- service::EdgeService::- findAllEdgesOfBuilding()	N.I

Test	Descrizione	Metodi	Stato
TU53	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto EdgeTable, sia possibile costruire un oggetto EdgeTable e inserirlo nel database locale.	model::dataaccess:- service::EdgeService:- convertAndInsert()	N.I
TU54	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto EdgeTypeTable, sia possibile costruire un oggetto EdgeTypeTable e inserirlo nel database locale.	model::dataaccess:- service::EdgeService:- convertAndInsertEdgeType()	N.I
TU55	Viene testato che sia possibile creare un oggetto ElevatorEdge dato un oggetto EdgeTable.	model::dataaccess:- service::EdgeService:- fromTableToBo()	N.I
TU56	Viene testato che sia possibile creare un oggetto StairEdge dato un oggetto EdgeTable.	model::dataaccess:- service::EdgeService:- fromTableToBo()	N.I
TU57	Viene testato che sia possibile creare un oggetto DefaultEdge dato un oggetto EdgeTable.	model::dataaccess:- service::EdgeService:- fromTableToBo()	N.I

Test	Descrizione	Metodi	Stato
TU58	Viene testato che sia possibile eliminare un PointOfInterest dal database locale, recuperarne uno o tutti quelli riguardanti un edificio, dato il major del suddetto edificio.	model::dataaccess:- service:- PointOfInterestService:- deletePointOfInterest() model::dataaccess:- service:- PointOfInterestService:- findPointOfInterest() model::dataaccess:- service:- PointOfInterestService:- findAllPointsWithMajor()	N.I
TU59	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto PointOfInterestTable, sia possibile costruire un oggetto PointOfInterestTable e inserirlo nel database locale.	model::dataaccess:- service:- PointOfInterestService:- convertAndInsert()	N.I
TU60	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto CategoryTable, sia possibile costruire un oggetto CategoryTable e inserirlo nel database locale.	model::dataaccess:- service:- PointOfInterestService:- convertAndInsertCategory()	N.I
TU61	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto RoiPoiTable, sia possibile costruire un oggetto RoiPoiTable e inserirlo nel database locale.	model::dataaccess:- service:- PointOfInterestService:- convertAndInsertRoiPoi()	N.I

Test	Descrizione	Metodi	Stato
TU62	Viene testato che sia possibile creare un oggetto PointOfInterestImp, dato un oggetto PointOfInterestTable.	model::dataaccess:- service:- PointOfInterestService:- fromTableToBo()	N.I
TU63	Viene testato che sia possibile eliminare una BuildingMap dal database locale, recuperarne una o tutte quelle presenti nel database locale.	model::dataaccess:- service::BuildingService:- deleteBuilding() model::dataaccess:- service::BuildingService:- findBuildingByMajor() model::dataaccess:- service::BuildingService:- findAllBuildings()	N.I
TU64	Viene testato che sia possibile recuperare una BuildingMap dal database remoto o le informazioni di tutte quelle presenti nel database remoto.	model::dataaccess:- service::BuildingService:- findAllRemoteBuildings() model::dataaccess:- service::BuildingService:- findRemoteBuildingByMajor()	N.I
TU65	Viene testato che sia possibile recuperare una BuildingMap dal database remoto e inserirla nel database locale.	model::dataaccess:- service::BuildingService:- retrieveAndInsertMap()	N.I
TU66	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto BuildingTable, sia possibile costruire un oggetto BuildingTable e inserirlo nel database locale.	model::dataaccess:- service::BuildingService:- convertAndInsert()	N.I

Test	Descrizione	Metodi	Stato
TU67	Viene testato che, dato il major di un edificio, sia possibile verificare la presenza della BuildingMap nel database locale, verificare se è aggiornata all'ultima versione disponibile e aggiornarla.	model::dataaccess:- service::BuildingService:- isBuildingMapPresent() model::dataaccess:- service::BuildingService:- isBuildingMapUpdated() model::dataaccess:- service::BuildingService:- updateBuildingMap()	N.I
TU68	Viene testato che sia possibile creare un oggetto BuildingMapImp, dato un oggetto BuildingTable.	model::dataaccess:- service::BuildingService:- fromTableToBo()	N.I
TU69	Viene testato che sia possibile creare e ritornare un oggetto SQLiteDaoFactory dato un oggetto SQLiteDatabase. Viene inoltre testato che sia possibile creare e ritornare un oggetto RemoteDaoFactory.	model::dataaccess::dao:- DaoFactoryHelper:- getSQLiteDaoFactory() model::dataaccess::dao:- DaoFactoryHelper:- getRemoteDaoFactory()	N.I

Test	Descrizione	Metodi	Stato
TU70	Viene testato che sia possibile creare e ritornare: un oggetto RemoteBuildingDao, un oggetto RemotePointOfInterestDao, un oggetto RemoteRegionOfInterestDao, un oggetto RemoteRoiPoiDao, un oggetto RemoteEdgeDao, un oggetto RemoteEdgeTypeDao, un oggetto RemoteCategoryDao, un oggetto RemotePhotoDao.	model::dataaccess::dao::-RemoteDaoFactory::-getBuildingDao() model::dataaccess::dao::-RemoteDaoFactory::-getPointOfInterestDao() model::dataaccess::dao::-RemoteDaoFactory::-getRoiPoiDao() model::dataaccess::dao::-RemoteDaoFactory::-getEdgeDao() model::dataaccess::dao::-RemoteDaoFactory::-getCategoryDao() model::dataaccess::dao::-RemoteDaoFactory::-getEdgeTypeDao() model::dataaccess::dao::-RemoteDaoFactory::-getPhotoDao() model::dataaccess::dao::-RemoteDaoFactory::-getRegionOfInterestDao()	N.I
TU71	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto BuildingTable, sia possibile costruire un oggetto BuildingTable e ritornarlo.	model::dataaccess::dao::-RemoteBuildingDao::-fromJSONToTable()	N.I

Test	Descrizione	Metodi	Stato
TU72	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto PointOfInterestTable, sia possibile costruire un oggetto PointOfInterestTable e ritornarlo.	model::dataaccess::dao::-RemotePointOfInterestDao::-fromJSONToTable()	N.I
TU73	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto RegionOfInterestTable, sia possibile costruire un oggetto RegionOfInterestTable e ritornarlo.	model::dataaccess::dao::-RemoteRegionOfInterestDao::-fromJSONToTable()	N.I
TU74	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto RoiPoiTable, sia possibile costruire un oggetto RoiPoiTable e ritornarlo.	model::dataaccess::dao::-RemoteRoiPoiDao::-fromJSONToTable()	N.I
TU75	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto EdgeTable, sia possibile costruire un oggetto EdgeTable e ritornarlo.	model::dataaccess::dao::-RemoteEdgeDao::-fromJSONToTable()	N.I
TU76	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto EdgeTypeTable, sia possibile costruire un oggetto EdgeTypeTable e ritornarlo.	model::dataaccess::dao::-RemoteEdgeTypeDao::-fromJSONToTable()	N.I

Test	Descrizione	Metodi	Stato
TU77	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto CategoryTable, sia possibile costruire un oggetto CategoryTable e ritornarlo.	model::dataaccess::dao::-RemoteCategoryDao::-fromJSONToTable()	N.I
TU78	Viene testato che, dato un oggetto JsonObject che possiede gli stessi valori di un oggetto PhotoTable, sia possibile costruire un oggetto PhotoTable e ritornarlo.	model::dataaccess::-dao::RemotePhotoDao::-fromJSONToTable()	N.I
TU79	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto BuildingTable.	model::dataaccess::dao::-BuildingTable::getId() model::dataaccess::dao::-BuildingTable::getUUID() model::dataaccess::dao::-BuildingTable::getMajor() model::dataaccess::-dao::BuildingTable::-getVersion() model::dataaccess::dao::-BuildingTable::getName() model::dataaccess::-dao::BuildingTable::-getDescription() model::dataaccess::-dao::BuildingTable::-getOpeningHours() model::dataaccess::-dao::BuildingTable::-getAddress() model::dataaccess::dao::-BuildingTable::getSize()	N.I

Test	Descrizione	Metodi	Stato
TU80	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto PointOfInterestTable.	model::dataaccess::dao::- PointOfInterestTable::- getId() model::dataaccess::dao::- PointOfInterestTable::- getName() model::dataaccess::dao::- PointOfInterestTable::- getDescription() model::dataaccess::dao::- PointOfInterestTable::- getCategoryId()	N.I
TU81	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto RegionOfInterestTable.	model::dataaccess::dao::- RegionOfInterestTable::- getId() model::dataaccess::dao::- RegionOfInterestTable::- getUUID() model::dataaccess::dao::- RegionOfInterestTable::- getMajor() model::dataaccess::dao::- RegionOfInterestTable::- getMinor()	N.I
TU82	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto RoiPoiTable.	model::dataaccess::dao::- RoiPoiTable::getRoiID() model::dataaccess::dao::- RoiPoiTable::getPoiID()	N.I

Test	Descrizione	Metodi	Stato
TU83	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto EdgeTable.	model::dataaccess::dao::- EdgeTable::getId() model::dataaccess::dao::- EdgeTable::getStartROI() model::dataaccess::dao::- EdgeTable::getEndROI() model::dataaccess::dao::- EdgeTable::getDistance() model::dataaccess::- dao::EdgeTable::- getCoordinate() model::dataaccess::dao::- EdgeTable::getTypeId() model::dataaccess::dao::- EdgeTable::getAction() model::dataaccess::- dao::EdgeTable::- getLongDescription()	N.I
TU84	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto EdgeTypeTable.	model::dataaccess::dao::- EdgeTypeTable::getId() model::dataaccess::- dao::EdgeTypeTable::- getTypeName()	N.I
TU85	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto CategoryTable.	model::dataaccess::dao::- CategoryTable::getId() model::dataaccess::- dao::CategoryTable::- getDescription()	N.I
TU86	Viene testato che sia possibile accedere a tutte le informazioni relative ad un oggetto PhotoTable.	model::dataaccess::dao::- PhotoTable::getId() model::dataaccess::dao::- PhotoTable::getUrl() model::dataaccess::dao::- PhotoTable::getEdgeId()	N.I

Test	Descrizione	Metodi	Stato
TU87	Viene testato che sia possibile creare e ritornare: un oggetto SQLiteBuildingDao, un oggetto SQLitePointOfInterestDao, un oggetto SQLiteRegionOfInterestDao, un oggetto SQLiteRoiPoiDao, un oggetto SQLiteEdgeDao, un oggetto SQLiteEdgeTypeDao, un oggetto SQLiteCategoryDao, un oggetto SQLitePhotoDao.	model::dataaccess::dao::- SQLiteDaoFactory::- getBuildingDao() model::dataaccess::dao::- SQLiteDaoFactory::- getPointOfInterestDao() model::dataaccess::dao::- SQLiteDaoFactory::- getRegionOfInterestDao() model::dataaccess::dao::- SQLiteDaoFactory::- getRoiPoiDao() model::dataaccess::dao::- SQLiteDaoFactory::- getEdgeDao() model::dataaccess::dao::- SQLiteDaoFactory::- getCategoryDao() model::dataaccess::dao::- SQLiteDaoFactory::- getEdgeTypeDao() model::dataaccess::dao::- SQLiteDaoFactory::- getPhotoDao()	N.I

Test	Descrizione	Metodi	Stato
TU88	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "Building" del database locale. In particolare, nel caso della ricerca viene testato che sia possibile effettuare la sia tramite identificativo che tramite major dell'edificio e che sia possibile ottenere le informazioni di tutte le mappe presenti sul database locale.	model::dataaccess::dao::- SQLiteBuildingDao::- insertBuilding() model::dataaccess::dao::- SQLiteBuildingDao::- deleteBuilding() model::dataaccess::dao::- SQLiteBuildingDao::- findBuildingById() model::dataaccess::dao::- SQLiteBuildingDao::- findBuildingByMajor() model::dataaccess::dao::- SQLiteBuildingDao::- findAllBuildings() model::dataaccess::dao::- SQLiteBuildingDao::- updateBuilding()	N.I
TU89	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto BuildingTable.	model::dataaccess::dao::- SQLiteBuildingDao::- cursorToType()	N.I
TU90	Viene verificato che, dato il major di un edificio, sia possibile verificare se la sua mappa è presente nel database locale.	model::dataaccess::dao::- SQLiteBuildingDao::- isBuildingMapPresent()	N.I

Test	Descrizione	Metodi	Stato
TU91	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "POI" del database locale. In particolare, nel caso della ricerca viene testato che, dato il major di un edificio, sia possibile recuperare tutti gli oggetti PointOfInterestTable che rappresentano i PointOfInterest di quell'edificio.	model::dataaccess::dao:- SQLitePointOfInterestDao:- insertPointOfInterest() model::dataaccess::dao:- SQLitePointOfInterestDao:- deletePointOfInterest() model::dataaccess::dao:- SQLitePointOfInterestDao:- findPointOfInterest() model::dataaccess::dao:- SQLitePointOfInterestDao:- findAllPointsWithMajor() model::dataaccess::dao:- SQLitePointOfInterestDao:- updatePointOfInterest()	N.I
TU92	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto PointOfInterestTable.	model::dataaccess::dao:- SQLitePointOfInterestDao:- cursorToType()	N.I
TU93	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "ROI" del database locale. In particolare, nel caso della ricerca viene testato che, dato il major di un edificio, sia possibile recuperare tutti gli oggetti RegionOfInterestTable che rappresentano le RegionOfInterest di quell'edificio.	model::dataaccess::dao:- SQLiteRegionOfInterestDao:- insertRegionOfInterest() model::dataaccess::dao:- SQLiteRegionOfInterestDao:- deleteRegionOfInterest() model::dataaccess::dao:- SQLiteRegionOfInterestDao:- findRegionOfInterest() model::dataaccess::dao:- SQLiteRegionOfInterestDao:- findAllRegionsWithMajor() model::dataaccess::dao:- SQLiteRegionOfInterestDao:- updateRegionOfInterest()	N.I

Test	Descrizione	Metodi	Stato
TU94	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto RegionOfInterestTable.	model::dataaccess::dao:- SQLiteRegionOfInterestDao:- cursorToType()	N.I
TU95	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "ROI-POI" del database locale. In particolare, nel caso della ricerca o della rimozione viene testato che, dato l'identificativo di un PointOfInterest, sia possibile recuperare o eliminare tutti gli oggetti RegionOfInterestTable che rappresentano tutte le RegionOfInterest che lo contengono e viceversa.	model::dataaccess::dao:- SQLiteRoiPoiDao:- insertRoiPoi() model::dataaccess::dao:- SQLiteRoiPoiDao:- deleteRoiPoisWherePoi() model::dataaccess::dao:- SQLiteRoiPoiDao:- deleteRoiPoisWhereRoi() model::dataaccess::dao:- SQLiteRoiPoiDao:- findAllRegionsWithPoi() model::dataaccess::dao:- SQLiteRoiPoiDao:- findAllPointsWithRoi() model::dataaccess::dao:- SQLiteRoiPoiDao:- updateRoiPoi()	N.I
TU96	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto RoiPoiTable.	model::dataaccess::dao:- SQLiteRoiPoiDao:- cursorToType()	N.I

Test	Descrizione	Metodi	Stato
TU97	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "Edge" del database locale. In particolare, nel caso della ricerca viene testato che, dato il major di un edificio, sia possibile recuperare tutti gli oggetti EdgeTable che rappresentano gli Edge di quell'edificio.	model::dataaccess::dao::SQLiteEdgeDao::insertEdge() model::dataaccess::dao::SQLiteEdgeDao::deleteEdge() model::dataaccess::dao::SQLiteEdgeDao::findEdge() model::dataaccess::dao::SQLiteEdgeDao::findAllEdgesOfBuilding() model::dataaccess::dao::SQLiteEdgeDao::updateEdge()	N.I
TU98	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto EdgeTable.	model::dataaccess::dao::SQLiteEdgeDao::cursorToType()	N.I
TU99	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "Category" del database locale.	model::dataaccess::dao::SQLiteCategoryDao::insertCategory() model::dataaccess::dao::SQLiteCategoryDao::deleteCategory() model::dataaccess::dao::SQLiteCategoryDao::findCategory() model::dataaccess::dao::SQLiteCategoryDao::updateCategory()	N.I

Test	Descrizione	Metodi	Stato
TU100	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto CategoryTable.	model::dataaccess::dao:- SQLiteCategoryDao:- cursorToType()	N.I
TU101	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "EdgeType" del database locale.	model::dataaccess::dao:- SQLiteEdgeTypeDao:- insertEdgeType() model::dataaccess::dao:- SQLiteEdgeTypeDao:- deleteEdgeType() model::dataaccess::dao:- SQLiteEdgeTypeDao:- findEdgeType() model::dataaccess::dao:- SQLiteEdgeTypeDao:- updateEdgeType()	N.I
TU102	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto EdgeTypeTable.	model::dataaccess::dao:- SQLiteEdgeTypeDao:- cursorToType()	N.I

Test	Descrizione	Metodi	Stato
TU103	Viene testato che sia possibile effettuare le operazioni CRUD sulla tabella "Photo" del database locale. In particolare, nel caso della ricerca viene testato che, dato l'identificativo di un Edge, sia possibile recuperare tutti gli oggetti PhotoTable che rappresentano le foto di quell'Edge.	model::dataaccess:- dao::SQLitePhotoDao:- insertPhoto() model::dataaccess:- dao::SQLitePhotoDao:- deletePhoto() model::dataaccess:- dao::SQLitePhotoDao:- findPhoto() model::dataaccess:- dao::SQLitePhotoDao:- findAllPhotosOfEdge() model::dataaccess:- dao::SQLitePhotoDao:- updatePhoto()	N.I
TU104	Viene testato che, dato un oggetto di tipo Cursor che rappresenta il risultato della query sul database locale, sia possibile creare un oggetto PhotoTable.	model::dataaccess:- dao::SQLitePhotoDao:- cursorToType()	N.I
TU105	Viene testato che sia possibile effettuare le operazioni CRUD sull'intero database locale.	model::dataaccess::dao:- SQLDao::insert() model::dataaccess::dao:- SQLDao::delete() model::dataaccess::dao:- SQLDao::query() model::dataaccess::dao:- SQLDao::update() model::dataaccess::dao:- SQLDao::rawQuery()	N.I
TU106	Viene testato che sia possibile creare e ritornare un'istanza di MapsDbHelper.	model::dataaccess:- dao::MapsDbHelper:- getInstance()	N.I

Test	Descrizione	Metodi	Stato
TU107	Viene testato che vengano creati il database e le sue tabelle e che venga effettuata la popolazione iniziale delle tabelle.	model::dataaccess:- dao::MapsDbHelper:- onCreate()	N.I
TU108	Viene testato che venga aggiornato il database in seguito all'aggiunta o alla rimozione di una tabella.	model::dataaccess:- dao::MapsDbHelper:- onUpgrade()	N.I
TU109	Viene testato che sia possibile recuperare l'URL del database remoto.	model::dataaccess:- dao::MapsDbHelper:- getRemoteDatabaseURL()	N.I
TU110	Viene testato che Compass sia effettivamente un in ascolto dei sensori	model::compass:- Compass:- registerListener()	N.I
TU111	Viene testato che Compass possa terminare l'ascolto dei sensori	model::compass:- Compass:- unregisterListener()	N.I
TU112	Viene testato che il valore lastCoordinate nel tempo non cambi con i sensori spenti e cambi con i sensori attivi	model::compass:- Compass:- getLastCoordinate()	N.I
TU113	Viene verificato che sia possibile verificare la possibilità di utilizzare il bluetooth e se grazie al quale sia possibile ricevere il segnale dei beacon, leggendo I suoi dati	presenter::Checker:- isBluetoothEnabled() presenter::Checker:- isBeaconDetected() presenter::Checker:- isTheBuildingMapInLocalDB()	N.I
TU114	Viene verificato che sia possibile accedere ad un oggetto NavigationManager utilizzando la classe NavigationManagerPresenter	presenter:- NavigationManagerPresenter:- getNavigationManager()	N.I

Test	Descrizione	Metodi	Stato
TU115	Viene verificato che sia possibile accedere ad un oggetto InformationManager utilizzando la classe InformationManagerPresenter	presenter::- InformationManagerPresenter::- getInformationManager()	N.I
TU116	Viene verificato che sia possibile accedere ad un oggetto DatabaseService utilizzando la classe DatabaseServiceManager	presenter::- DatabaseServiceManager::- getDatabaseService()	N.I
TU117	Viene verificato che sia possibile accedere ad un oggetto Setting utilizzando la classe SettingManger	presenter::- SettingManager::- getSetting()	N.I
TU118	Viene verificato che sia possibile recuperare le informazioni di un ContentProvider per effettuare una ricerca utilizzando la classe SearchSuggestionProvider	presenter::- SearchSuggestionsProvider::- query() presenter::- SearchSuggestionsProvider::- getType() presenter::- SearchSuggestionsProvider::- insert() presenter::- SearchSuggestionsProvider::- update() presenter::- SearchSuggestionsProvider::- onCreate()	N.I
TU119	Viene verificato che sia possibile recuperare il nome dell'edificio da InformationManager e che tale nome sia passato ad HomeView	presenter::-HomeActivity::- updateBuildingName()	N.I

Test	Descrizione	Metodi	Stato
TU120	Viene verificato che sia possibile recuperare la descrizione dell'edificio da InformationManager e che tale descrizione sia passata ad HomeView	presenter::HomeActivity::updateBuildingDescription()	N.I
TU121	Viene verificato che sia possibile recuperare le ore di apertura dell'edificio da InformationManager e che tale orario sia passato ad HomeView	presenter::HomeActivity::updateBuildingOpeningHours()	N.I
TU122	Viene verificato che sia possibile recuperare le categorie di POI dell'edificio da InformationManager e che tali categorie siano passate ad HomeView	presenter::HomeActivity::updatePoiCategoryList()	N.I
TU123	Viene verificato che sia possibile recuperare l'indirizzo dell'edificio da InformationManager e che tale indirizzo sia passato ad HomeView	presenter::HomeActivity::updateBuildingAddress()	N.I
TU124	Viene verificato che sia possibile recuperare i nomi dei POI dell'edificio da InformationManager	presenter::HomeActivity::enableSuggestions()	N.I
TU125	Viene verificato che sia possibile mostrare le categorie di POI	presenter::HomeActivity::showPoisCategory()	N.I
TU126	Viene verificato che sia possibile mostrare le preferenze utente	presenter::HomeActivity::showPreferences()	N.I

Test	Descrizione	Metodi	Stato
TU127	Viene verificato che sia possibile mostrare la guida dell'applicazione	presenter::HomeActivity::showHelp()	N.I
TU128	Viene verificato che sia possibile avviare la navigazione	presenter::HomeActivity::startNavigation()	N.I
TU129	Viene verificato che sia possibile mostrare le mappe salvate nel database locale	presenter::HomeActivity::showLocalMaps()	N.I
TU130	Viene verificato che sia possibile far partire la navigazione utilizzando l'identificativo di un POI appartenente ad una certa categoria	presenter::PoiCategoryActivity::startNavigation()	N.I
TU131	Viene verificato che sia possibile gestire le informazioni di navigazione	presenter::NavigationActivity::pathError() presenter::NavigationActivity::informationUpdate()	N.I
TU132	Viene verificato che sia possibile visualizzare le informazioni dettagliate di navigazione	presenter::NavigationActivity::showDetailedInformation()	N.I
TU133	Viene verificato che sia possibile gestire un'immagine utilizzando la classe ImageAdapter	presenter::ImageAdapter::getCount() presenter::ImageAdapter::getItem() presenter::ImageAdapter::getView()	N.I

Test	Descrizione	Metodi	Stato
TU134	Viene verificato che sia possibile gestire un insieme di immagini utilizzando la classe ImageListFragment	presenter::- ImageListFragment::- onItemClick() presenter::- ImageListFragment::- newInstance() presenter::- ImageListFragment::- onCreateView()	N.I
TU135	Viene verificato che sia possibile gestire le opzioni sviluppatore utilizzando la classe MainDeveloperPresenter	presenter::- MainDeveloperPresenter::- isDeveloper() presenter::- MainDeveloperPresenter::- startDeveloperUnlocker() presenter::- MainDeveloperPresenter::- startDeveloperOptions()	N.I
TU136	Viene verificato che sia possibile gestire lo sblocco delle opzioni sviluppatore utilizzando la classe DeveloperUnlockerActivity	presenter::- DeveloperUnlockerActivity::- unlockDeveloper()	N.I
TU137	Viene verificato che sia possibile avviare un nuovo log e accedere ai log salvati sul dispositivo utilizzando la classe MainDeveloperActivity	presenter::- MainDeveloperActivity::- showDetailedLog() presenter::- MainDeveloperActivity::- startNewLog()	N.I
TU138	Viene verificato che sia possibile salvare le preferenze utente utilizzando la classe PreferencesActivity	presenter::- PreferencesActivity::- savePreferences()	N.I

Test	Descrizione	Metodi	Stato
TU139	Viene verificato che vengano visualizzate le mappe locali terminato il download di una mappa utilizzando la classe MapDownloaderActivity	presenter::- MapDownloaderActivity::- downloadFinished()	N.I
TU140	Viene verificato che sia possibile effettuare il download di una mappa utilizzando la classe RemoteMapManager	presenter::- RemoteMapManagerActivity::- downloadMap()	N.I
TU141	Viene verificato che sia possibile aggiornare o rimuovere una mappa già presente sul dispositivo utilizzando la classe LocalMapActivity utilizzando LocalMapActivity	presenter::- LocalMapActivity::- updateMap() presenter::- LocalMapActivity::- deleteMap()	N.I
TU142	Viene verificato che venga visualizzata la guida dell'applicativo utilizzando HelpActivity	presenter::HelpActivity::- onCreate()	N.I
TU143	Viene verificato che venga visualizzata la foto selezionata dall'utente utilizzando ImageDetailActivity	presenter::- ImageDetailActivity::- onCreate()	N.I
TU144	Viene verificato che venga visualizzata la lista di POI rilevati dal dispositivo utilizzando la classe NearbyPoiActivity	presenter::- NearbyPoiActivity::- onCreate()	N.I

Test	Descrizione	Metodi	Stato
TU145	Viene verificato che sia possibile gestire le foto e la descrizione dettagliata relativa ad una certa istruzione utilizzando la classe DetailedInformationActivity	presenter::- DetailedInformationActivity::- updatePhoto() presenter::- DetailedInformationActivity::- updateDetailedDescription()	N.I
TU146	Viene verificato che sia possibile gestire la lista di anteprime delle immagini di un certo POI utilizzando la classe ImagePagerAdapter	presenter::- ImagePagerAdapter::- getCount() presenter::- ImagePagerAdapter::- getItem()	N.I
TU147	Viene verificato che sia possibile gestire la lista di immagini di un certo POI utilizzando la classe ImageAdapter	presenter::ImageAdapter::- getCount() presenter::ImageAdapter::- getItem() presenter::ImageAdapter::- getItemId() presenter::ImageAdapter::- getView()	N.I
TU148	Viene verificato che venga visualizzata la schermata iniziale dell'applicazione utilizzando la classe MainActivity	presenter::MainActivity::- onCreate()	N.I
TU149	Viene verificato che venga arrestata l'attività di log e che venga salvato il log utilizzando la classe LogginActivity	presenter::- LoggingActivity::- stopLogging()	N.I
TU150	Viene verificato che sia possibile eliminare un log salvato utilizzando la classe LogInformationActivity	presenter::- LogInformationActivity::- deleteLog()	N.I

Test	Descrizione	Metodi	Stato
TU151	Viene verificato che sia possibile gestire la lista di indicazioni utili per raggiungere una certa destinazione utilizzando la classe NavigationAdapter	presenter::- NavigationAdapter::- getCount() presenter::- NavigationAdapter::- getItem() presenter::- NavigationAdapter::- getItemId() presenter::- NavigationAdapter::- getView()	N.I
TU152	Viene verificato che vengano visualizzate le informazioni di un edificio	view::HomeViewImp::- setBuildingName() view::HomeViewImp::- setBuildingOpeningHours() view::HomeViewImp::- setPoiCategoryListAdapter() view::HomeViewImp::- setBuildingAddress()	N.I
TU153	Viene verificato che sia possibile visualizzare le istruzioni di navigazione	view::- NavigationViewImp::- setInstructionAdapter() view::- NavigationViewImp::- refreshInstructions()	N.I
TU154	Viene verificato che vengano visualizzate le categorie dei POI	view::- PoiCategoryViewImp::- setPoiListAdapter()	N.I
TU155	Viene verificato che vengano visualizzate le indicazioni dettagliate e le foto associate ad un arco	view::- DetailedInformationViewImp::- setPhotos() view::- DetailedInformationViewImp::- setDetailedDescription()	N.I

Test	Descrizione	Metodi	Stato
TU156	Viene verificato che vengano visualizzati tutti i POI associati ad un certo ROI	view::- NearbyPoiViewImp::- setAdapter()	N.I
TU157	Viene verificato che sia possibile visualizzare una immagine	view::- ImageDetailViewImp::- setAdapter()	N.I
TU158	Viene verificato che sia possibile visualizzare la guida	view::HelpViewImp::- setHelp()	N.I
TU159	Viene verificato che siano visualizzate le impostazioni per gestire le mappe locali	view::- LocalMapManagerViewImp::- refreshMaps() view::- LocalMapManagerViewImp::- setAdapter()	N.I
TU160	Viene verificato che siano visualizzate le impostazioni per gestire le mappe remote	view::- RemoteMapManagerViewImp::- setRemoteMaps()	N.I
TU161	Viene verificato che sia visualizzata la schermata di download di una mappa	view::- MapDownloaderView::- setDownloadingMap() view::- MapDownloaderView::- setProgressDownload()	N.I
TU162	Viene verificato che siano visualizzate le impostazioni per gestire le preferenze riguardanti l'applicazione	view::- PreferencesViewImp::- setPathPreferences() view::- PreferencesViewImp::- setInstructionPreferences()	N.I

Test	Descrizione	Metodi	Stato
TU163	Viene verificato che sia possibile inserire un codice per sbloccare le funzionalità sviluppatore e visualizzare un messaggio di errore in caso di codice errato	view::- DeveloperUnlockerViewImp::- showWrongCode()	N.I
TU164	Viene verificato che vengano visualizzati i log salvati	view::- MainDeveloperViewImp::- setLogsAdapter()	N.I
TU165	Viene verificato che vengano visualizzati gli identificati dei beacon circostanti	view::LoggingViewImp::- setBeaconListAdapter()	N.I
TU166	Viene verificato che vengano visualizzato il dettaglio di un log e che questo possa essere eliminato	view::- LogInformationViewImp::- setBeaconAdapter()	N.I
TU167	Viene verificato che venga visualizzata la lista di anteprime delle immagini relative ad una certa istruzione di navigazione	view::- ImageListFragmentViewImp::- setImageAdapter()	N.I
TU168	Viene verificato che venga visualizzata la schermata di caricamento dell'applicazione	view::MainView::- setLoadingView()	N.I

Tabella 6: Tabella descrizione test unità

E Resoconto delle attività di verifica - fase A

All'interno di questa prima fase_g, secondo quanto riportato nel documento *Piano di progetto*, 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 tratte 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*. 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 7: Errori trovati tramite verifica manuale dei documenti durante la fase 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 8: Nuovi termini da inserire nel *Glossario* individuati tramite verifica manuale dei documenti durante la fase 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*.

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

Tabella 9: Errori trovati tramite verifica automatica dei documenti durante la fase 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 10: Esiti del calcolo dell'indice di leggibilità effettuato tramite strumenti automatici durante la fase 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) sono state rese disponibili 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). Possiamo quindi 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 3.2 "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 il processo_g di verifica, 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_g 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 [3.2](#) "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).

F Resoconto delle attività di verifica - fase AD

All'interno di questa fase_g, secondo quanto riportato nel documento *Piano di progetto*, 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 tratte anche delle conclusioni sui risultati ottenuti e su come essi possono essere migliorati.

F.1 Verifica sui processi

F.1.1 processo di documentazione

F.1.1.1 Miglioramento costante All'inizio della fase_g AD il processo_g di documentazione si posizionava al livello 2 della scala CMM_g.

In seguito alla riorganizzazione del documento *Norme di progetto* e grazie ad una maggiore esperienza dei membri del gruppo, i processi e la loro organizzazione sono notevolmente migliorati. Ciò ha permesso di raggiungere il terzo livello CMM_g.

In questa fase_g abbiamo, inoltre, iniziato a misurare la qualità dei processi ampliando le metriche utilizzate e gli obiettivi di qualità scelti, fissandoli in modo quantitativo.

F.1.1.2 Rispetto della pianificazione Per capire se le attività di un processo_g sono 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.

Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di stesura di ogni documento nella fase_g AD:

Documento	Schedule Variance	Esito
<i>Piano di progetto v2.00</i>	-43%	Ottimale
<i>Norme di progetto v2.00</i>	-26%	Ottimale
<i>Analisi dei requisiti v2.00</i>	0%	Ottimale
<i>Piano di qualifica v2.00</i>	0%	Ottimale
<i>Glossario v2.00</i>	0%	Ottimale

Tabella 11: Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase AD

F.1.1.3 Rispetto del budget Per capire se i costi di un processo_g rientrano 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. Di seguito sono riportati i valori ottenuti calcolando la Budget Variance sui tempi di stesura di ogni documento nella fase_g AD:

Documento	Budget Variance	Esito
<i>Piano di progetto v2.00</i>	0%	Ottimale
<i>Norme di progetto v2.00</i>	+14%	Non accettabile
<i>Analisi dei requisiti v2.00</i>	+50%	Non accettabile
<i>Piano di qualifica v2.00</i>	+14%	Non accettabile
<i>Glossario v2.00</i>	0%	Ottimale
Totale processo _g di documentazione	+32%	Non accettabile

Tabella 12: Esiti del calcolo della Budget Variance sul processo di documentazione durante la fase AD

Si può notare dalla tabella che per il documento *Analisi dei requisiti v2.00* è stato investito un notevole numero di risorse in più rispetto a quanto preventivato: ciò è dovuto al fatto che nella fase_g corrente sono stati individuati molti nuovi requisiti, oltre alle correzioni che sono state effettuate in seguito alle osservazioni e segnalazioni del committente e del proponente. Oltre a ciò, si può notare che anche i documenti *Norme di progetto v2.00* e

Piano di qualifica v2.00 hanno richiesto più risorse di quanto preventivato: ciò è dovuto al fatto che erano necessarie molte correzioni ai documenti in questione e, in alcuni sezioni, addirittura una completa ristrutturazione.

F.1.2 processo di verifica

F.1.2.1 Miglioramento costante Nonostante l'adozione di nuove metriche e una maggiore regolamentazione del processo_g di verifica, il team_g non è ancora in grado di individuare miglioramenti tali da raggiungere il terzo livello CMM_g, pertanto il processo_g di verifica rimane al secondo livello (Ripetibile).

F.1.2.2 Rispetto della pianificazione Per capire se le attività di un processo_g sono 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. Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di verifica nella fase_g AD:

Processo	Schedule Variance	Esito
Processo _g di verifica	-15%	Ottimale

Tabella 13: Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase AD

F.1.2.3 Rispetto del budget Per il processo_g di verifica è stato investito un minor numero di risorse rispetto a quanto preventivato, di conseguenza il valore della Budget Variance risulta **ottimale**. Di seguito sono riportati i valori ottenuti:

Processo	Budget Variance	Esito
Processo _g di verifica	-29%	Ottimale

Tabella 14: Esiti del calcolo della Budget Variance sul processo di verifica durante la fase AD

F.2 Verifica sui prodotti

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

F.2.1 Documenti

In questa sezione vengono riportati gli esiti delle attività di verifica svolte sui documenti.

Tali esiti sono strettamente correlati agli obiettivi di qualità dei documenti enunciati alla sezione 2.1 del presente documento.

F.2.1.1 Leggibilità e comprensibilità 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.

Il documento *Glossario v2.00* ha dato esito **ottimale**, mentre tutti gli altri documenti prodotti hanno dato esito **accettabile**.

F.2.1.2 Correttezza ortografica 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.

Di seguito sono riportati gli errori ortografici trovati tramite verifica automatica dei documenti durante la fase_g AD.

Errori ortografici	11
--------------------	----

Tabella 15: Errori ortografici trovati tramite verifica automatica dei documenti durante la fase AD

Tutti gli errori ortografici rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.

F.2.1.3 Correttezza concettuale 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 correggere tutti gli errori trovati.

Di seguito sono riportati gli errori concettuali trovati dei documenti durante la fase_g AD.

Errori concettuali	5
--------------------	---

Tabella 16: Errori concettuali trovati tramite verifica manuale dei documenti durante la fase AD

Tutti gli errori concettuali rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.

G Resoconto delle attività di verifica - fase PA

All'interno di questa fase_g, secondo quanto riportato nel documento *Piano di progetto*, 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 tratte anche delle conclusioni sui risultati ottenuti e su come essi possono essere migliorati.

G.1 Verifica sui processi

G.1.1 processo di documentazione

G.1.1.1 Miglioramento costante All'inizio della fase_g PA il processo_g di documentazione si posizionava al livello 3 della scala CMM_g.

A causa del carico di lavoro richiesto dagli altri documenti, il gruppo non è riuscito a definire nuove norme e ricercare nuovi strumenti per permettere il passaggio al livello 4 della scala CMM_g.

G.1.1.2 Rispetto della pianificazione Per capire se le attività di un processo_g sono 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.

Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di stesura di ogni documento nella fase_g PA:

Documento	Schedule Variance	Esito
<i>Piano di progetto v3.00</i>	0%	Ottimale
<i>Norme di progetto v3.00</i>	+33%	Non accettabile
<i>Analisi dei requisiti v3.00</i>	+25%	Non accettabile
<i>Piano di qualifica v3.00</i>	0%	Ottimale
<i>Glossario v3.00</i>	-6%	Ottimale
<i>Specifica tecnica v1.00</i>	+11%	Non accettabile

Tabella 17: Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase PA

Come è possibile osservare dai dati della tabella, c'è stata una sottostima dei giorni necessari a completare i documenti *Norme di progetto* e *Analisi dei requisiti*: ciò è dovuto principalmente al ritardo accumulato nella fase_g precedente, che ha ritardato la data di inizio delle attività di questa fase_g e di conseguenza influito negativamente sulla Schedule Variance.

G.1.1.3 Rispetto del budget Per capire se i costi di un processo_g rientrano 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. Di seguito sono riportati i valori ottenuti calcolando la Budget Variance sui tempi di stesura di ogni documento nella fase_g PA:

Documento	Budget Variance	Esito
<i>Piano di progetto v3.00</i>	-10%	Ottimale
<i>Norme di progetto v3.00</i>	-6%	Ottimale
<i>Analisi dei requisiti v3.00</i>	-14%	Ottimale
<i>Piano di qualifica v3.00</i>	-22%	Ottimale
<i>Glossario v3.00</i>	0%	Ottimale
<i>Specifiche tecniche v1.00</i>	+16%	Non Accettabile
Totale processo _g di documentazione	-5%	Ottimale

Tabella 18: Esiti del calcolo della Budget Variance sul processo di documentazione durante la fase PA

Si può notare dalla tabella che il documento *Specifiche tecniche* ha richiesto più ore da *Progettista* di quante ne fossero state preventivate, mentre per i restanti documenti c'è stata una sovrastima di ore per i rispettivi ruoli.

G.1.2 processo di verifica

G.1.2.1 Miglioramento costante Il processo_g di verifica inizia ad essere ben documentato e ben gestito, inoltre ogni membro del gruppo ha ormai chiare in mente le procedure di verifica e conosce a fondo le *Norme di progetto*: tutto ciò ha permesso di raggiungere il terzo livello della scala CMM_g (Defined).

G.1.2.2 Rispetto della pianificazione Per capire se le attività di un processo_g sono 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.

Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di verifica nella fase_g PA:

Processo	Schedule Variance	Esito
Processo _g di verifica	-42%	Ottimale

Tabella 19: Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase PA

G.1.2.3 Rispetto del budget Per il processo_g di verifica è stato investito un minor numero di risorse rispetto a quanto preventivato, di conseguenza il valore della Budget Variance risulta **ottimale**.

Di seguito sono riportati i valori ottenuti:

Processo	Budget Variance	Esito
Processo _g di verifica	-22%	Ottimale

Tabella 20: Esiti del calcolo della Budget Variance sul processo di verifica durante la fase PA

G.2 Verifica sui prodotti

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

G.2.1 Documenti

In questa sezione vengono riportati gli esiti delle attività di verifica svolte sui documenti.

Tali esiti sono strettamente correlati agli obiettivi di qualità dei documenti enunciati alla sezione [2.1](#) del presente documento.

G.2.1.1 Leggibilità e comprensibilità 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.

Il documento *Glossario v3.00* ha dato esito **ottimale**, mentre tutti gli altri documenti prodotti hanno dato esito **accettabile**.

G.2.1.2 Correttezza ortografica 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. Di seguito sono riportati gli errori ortografici trovati tramite verifica automatica dei documenti durante la fase_g PA.

Errori ortografici	7
--------------------	---

Tabella 21: Errori ortografici trovati tramite verifica automatica dei documenti durante la fase PA

Tutti gli errori ortografici rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.

G.2.1.3 Correttezza concettuale 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 correggere tutti gli errori trovati.

Di seguito sono riportati gli errori concettuali trovati dei documenti durante la fase_g PA.

Errori concettuali	9
--------------------	---

Tabella 22: Errori concettuali trovati tramite verifica manuale dei documenti durante la fase PA

Tutti gli errori concettuali rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.

H Resoconto delle attività di verifica - fase PDROB

All'interno di questa fase_g, secondo quanto riportato nel documento *Piano di progetto*, 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 tratte anche delle conclusioni sui risultati ottenuti e su come essi possono essere migliorati.

H.1 Verifica sui processi

H.1.1 processo di documentazione

H.1.1.1 Miglioramento costante All'inizio della fase_g PDROB il processo_g di documentazione si posizionava al livello 3 della scala CMM_g.

A causa del carico di lavoro richiesto dagli altri documenti, il gruppo non è riuscito a definire nuove norme e ricercare nuovi strumenti tanto da permettere il passaggio al livello 4 della scala CMM_g.

H.1.1.2 Rispetto della pianificazione Per capire se le attività di un processo_g sono 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.

Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di stesura di ogni documento nella fase_g PDROB:

Documento	Schedule Variance	Esito
<i>Piano di progetto v4.00</i>	0%	Ottimale
<i>Norme di progetto v4.00</i>	0%	Ottimale
<i>Analisi dei requisiti v4.00</i>	0%	Ottimale
<i>Piano di qualifica v4.00</i>	0%	Ottimale
<i>Glossario v4.00</i>	0%	Ottimale
<i>Definizione di prodotto v1.00</i>	+42%	Non accettabile

Tabella 23: Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase PDROB

Come è possibile osservare dai dati della tabella, c'è stata una sottostima dei giorni necessari a completare il documento *Definizione di prodotto*: ciò è dovuto principalmente al ritardo accumulato nella fase_g precedente, che ha ritardato la data di inizio dell'attività di stesura di questo documento e di conseguenza ha influito negativamente sulla Schedule Variance.

H.1.1.3 Rispetto del budget Per capire se i costi di un processo_g rientrano 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. Di seguito sono riportati i valori ottenuti calcolando la Budget Variance sui tempi di stesura di ogni documento nella fase_g PDROB:

Documento	Budget Variance	Esito
<i>Piano di progetto v4.00</i>	-40%	Ottimale
<i>Norme di progetto v4.00</i>	0%	Ottimale
<i>Analisi dei requisiti v4.00</i>	-63%	Ottimale
<i>Piano di qualifica v4.00</i>	-22%	Ottimale
<i>Glossario v4.00</i>	0%	Ottimale
<i>Definizione di prodotto v1.00</i>	+58%	Non Accettabile
Totale processo _g di documentazione	4%	Accettabile

Tabella 24: Esiti del calcolo della Budget Variance sul processo di documentazione durante la fase PDROB

Nonostante il documento *Definizione di prodotto* abbia richiesto più ore di quante ne fossero state preventivate, il calcolo della Budget Variance sull'intero processo_g di documentazione ha dato esito **accettabile**.

H.1.2 processo di verifica

H.1.2.1 Miglioramento costante Il gruppo non ha rilevato miglioramenti tali da raggiungere il quarto livello CMM_g, pertanto il processo_g di verifica rimane al terzo livello (Defined).

H.1.2.2 Rispetto della pianificazione Per capire se le attività di un processo_g sono 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.

Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di verifica nella fase_g PDROB:

Processo	Schedule Variance	Esito
Processo _g di verifica	+4%	Accettabile

Tabella 25: Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase PDROB

H.1.2.3 Rispetto del budget Per il processo_g di verifica è stato investito un minor numero di risorse rispetto a quanto preventivato, di conseguenza il valore della Budget Variance risulta **ottimale**.

Di seguito sono riportati i valori ottenuti:

Processo	Budget Variance	Esito
Processo _g di verifica	-10%	Ottimale

Tabella 26: Esiti del calcolo della Budget Variance sul processo di verifica durante la fase PDROB

H.2 Verifica sui prodotti

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

H.2.1 Documenti

In questa sezione vengono riportati gli esiti delle attività di verifica svolte sui documenti.

Tali esiti sono strettamente correlati agli obiettivi di qualità dei documenti enunciati alla sezione [2.1](#) del presente documento.

H.2.1.1 Leggibilità e comprensibilità 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.

Il documento *Glossario v4.00* ha dato esito **ottimale**, mentre tutti gli altri documenti prodotti hanno dato esito **accettabile**.

H.2.1.2 Correttezza ortografica 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. Di seguito sono riportati gli errori ortografici trovati tramite verifica automatica dei documenti durante la fase_g PDROB.

Errori ortografici	8
--------------------	---

Tabella 27: Errori ortografici trovati tramite verifica automatica dei documenti durante la fase PDROB

Tutti gli errori ortografici rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.

H.2.1.3 Correttezza concettuale 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 correggere tutti gli errori trovati.

Di seguito sono riportati gli errori concettuali trovati dei documenti durante la fase_g PDROB.

Errori concettuali	2
--------------------	---

Tabella 28: Errori concettuali trovati tramite verifica manuale dei documenti durante la fase PDROB

Tutti gli errori concettuali rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.