



CLIPS

Piano di Qualifica v4.0.0

Sommario

Questo documento ha lo scopo di fissare le norme necessarie ad assicurare i requisiti qualitativi del *progetto*_g CLIPS, regolamentando le operazioni di pianificazione e di verifica attuate per rispettare tali norme.

Nome del documento	Piano di Qualifica
Versione	4.0.0
Data di redazione	2016-09-10
Redazione	Andrea Grendene Tommaso Panozzo
Verifica	Luca Soldera
Approvazione	Andrea Grendene
Uso	Esterno
Lista di distribuzione	prof. Tullio Vardanega prof. Riccardo Cardin Miriade SpA

Diario delle modifiche

Versione	Riepilogo	Autore	Ruolo	Data
4.0.0	Approvazione documento	Andrea Grendene	Responsabile	2016-09-11
3.1.0	Verifica documento	Luca Soldera	Verificatore	2016-09-11
3.0.2	Completata la sezione dell'esito della Revisione di Qualifica . Aggiornato il numero di attributi nelle metriche delle classi	Andrea Grendene	Progettista	2016-09-10
3.0.1	Aggiunte metriche documenti e processi per il Periodo di Codifica dei Requisiti Desiderabili e Opzionali e il Periodo di Validazione e Collaudo . Spostato in una sezione a sè le metriche sul codice . Aggiunto bozza dell'esito della Revisione di Qualifica	Andrea Grendene	Progettista	2016-09-09
3.0.0	Approvazione documento	Tommaso Panozzo	Responsabile	2016-08-16
2.3.0	Verifica documento	Viviana Alessio	Verificatore	2016-08-16
2.2.4	Aggiunti esiti dei primi test di qualità Esiti test di qualità	Andrea Grendene	Progettista	2016-08-15
2.2.3	Aggiunti test di unità Test di unità	Andrea Grendene	Progettista	2016-08-15
2.2.2	Aggiunti Schedule Variance, Budget Variance e indice Gulpease del Periodo di Progettazione di Dettaglio e Codifica Verifica periodo di Progettazione di Dettaglio e Codifica	Andrea Grendene	Progettista	2016-08-15
2.2.1	Aggiunti test di validazione Test di validazione	Luca Soldera	Progettista	2016-08-15
2.2.0	Aggiunto modello CMM in metriche per i processi	Luca Soldera	Progettista	2016-07-14
2.1.0	Verifica documento	Matteo Franco	Verificatore	2016-07-13
2.0.2	Aggiunti test di validazione Test di validazione	Luca Soldera	Progettista	2016-07-13
2.0.1	Aggiunti test di validazione Test di validazione	Luca Soldera	Progettista	2016-07-02

Versione	Riepilogo	Autore	Ruolo	Data
2.0.0	Approvazione del documento	Matteo Franco	Responsabile	2016-06-10
1.3.0	Verifica generale	Viviana Alessio	Verificatore	2016-06-08
1.2.3	Aggiunti i risultati di Schedule Variance, Budget Variance e indice Gulpease della sezione Periodo di Progettazione Architettuale	Tommaso Panozzo	Analista	2016-05-09
1.2.2	Aggiunti i test d'integrazione nella sezione Test di integrazione	Andrea Grendene	Progettista	2016-05-31
1.2.1	Aggiunti i test di validazione nella sezione Test di validazione e i test di sistema nella sezione Test di sistema	Andrea Grendene	Progettista	2016-05-30
1.2.0	Verifica delle modifiche	Enrico Bellio	Verificatore	2016-05-22
1.1.2	Aggiornata la sezione Portabilità in seguito alla riunione del 2016-05-11	Tommaso Panozzo	Analista	2016-05-12
1.1.1	Corretti gli errori risultati dalla verifica	Andrea Grendene	Progettista	2016-05-08
1.1.0	Verifica generale	Enrico Bellio	Verificatore	2016-05-07
1.0.5	Aggiornate le sezioni Resoconto dell'attività di verifica e Esito delle revisioni	Tommaso Panozzo	Analista	2016-05-03
1.0.4	Aggiunte le sezioni Schedule Variance , Budget Variance , Copertura del codice , Numero di linee per metodo e l'Esito delle revisioni . Modificate la tabella della Riepilogo	Tommaso Panozzo	Analista	2016-04-28
1.0.3	Eliminata la sezione 3.7 Strumenti. Spostata la sezione 3.7.6 nella sezione Modello a V	Andrea Grendene	Progettista	2016-04-27
1.0.2	Modificate le sezioni 3.8.2.1, 3.8.2.2, 3.8.2.3, 3.8.2.4, 3.8.2.5 e aggiunte le sottosezioni 3.8.2.6 e 3.8.2.7. Stesa la prima parte della Specifica dei test	Andrea Grendene	Progettista	2016-04-26

Versione	Riepilogo	Autore	Ruolo	Data
1.0.1	Piccole modifiche e correzioni grammaticali nelle sezioni Portabilità, Procedure di controllo di qualità di processo , Organizzazione , Misure e metriche , e Comunicazione e risoluzione delle anomalie	Tommaso Panozzo	Analista	2016-04-24
1.0.0	Approvazione documento	Viviana Alessio	Responsabile	2016-04-06
0.8.0	Terminata la verifica delle modifiche	Tommaso Panozzo	Verificatore	2016-04-05
0.7.1	Aggiunti i valori dell'indice <i>Gulpease_g</i> dei documenti e il loro esito	Andrea Grendene	Analista	2016-04-05
0.7.0	Applicate le modifiche proposte dal Verificatore	Andrea Grendene	Analista	2016-04-05
0.6.0	Terminata la verifica del documento	Tommaso Panozzo	Verificatore	2016-04-04
0.5.1	Terminata stesura della struttura del <i>Resoconto dell'attività di verifica</i> (senza i valori dell'indice <i>Gulpease_g</i> nella tabella)	Andrea Grendene	Analista	2016-03-25
0.5.0	Terminata stesura della sezione <i>Gestione amministrativa della revisione</i>	Andrea Grendene	Analista	2016-03-25
0.4.0	Terminata stesura della sezione <i>Visione generale della strategia</i>	Andrea Grendene	Analista	2016-03-25
0.3.2	Terminata stesura della sezione <i>Tecniche di analisi</i>	Andrea Grendene	Analista	2016-03-24
0.3.1	Terminata stesura delle sezioni <i>Risorse e Misure e Metriche</i>	Andrea Grendene	Analista	2016-03-23
0.3.0	Terminata stesura delle sezioni da <i>Procedure di controllo di qualità di processo a Responsabilità</i>	Andrea Grendene	Analista	2016-03-22
0.2.2	Terminata stesura della sezione <i>Definizione degli obiettivi di qualità</i>	Andrea Grendene	Analista	2016-03-21
0.2.1	Terminata stesura dell' <i>Introduzione</i>	Andrea Grendene	Analista	2016-03-20
0.2.0	Stesura della prima parte dei <i>Riferimenti</i>	Andrea Grendene	Analista	2016-03-19

Versione	Riepilogo	Autore	Ruolo	Data
0.1.1	Stesura dell'introduzione fino ai Riferimenti	Andrea Grendene	Analista	2016-03-18
0.1.0	Impostazione della struttura e dei dettagli del documento	Andrea Grendene	Analista	2016-03-15

Indice

1	Introduzione	9
1.1	Scopo del documento	9
1.2	Scopo del <i>prodotto_g</i>	9
1.3	Glossario	9
1.4	Riferimenti	9
1.4.1	Normativi	9
1.4.2	Informativi	9
2	Definizione obiettivi di qualità	10
2.1	Qualità di prodotto	10
2.1.1	Funzionalità	10
2.1.2	Affidabilità	10
2.2	Usabilità	10
2.3	Efficienza	11
2.4	Manutenibilità	11
2.5	Portabilità	11
2.6	Altre qualità	11
3	Visione generale della strategia	12
3.1	Procedure di controllo di qualità di processo	12
3.2	Procedure di controllo di qualità di <i>prodotto_g</i>	12
3.3	Organizzazione	12
3.4	Pianificazione strategica e temporale	13
3.5	Responsabilità	13
3.6	Risorse	14
3.6.1	Necessarie	14
3.6.2	Disponibili	14
3.7	Misure e metriche	14
3.7.1	Metriche per i processi	15
3.7.1.1	Modello <i>CMM_g</i>	15
3.7.1.2	Schedule Variance	15
3.7.1.3	Budget Variance	15
3.7.2	Metriche per i documenti	16
3.7.2.1	Indice <i>Gulpease_g</i>	16
3.7.3	Metriche per il codice	16
3.7.3.1	Numero di parametri	16
3.7.3.2	Complessità ciclomatica	16
3.7.3.3	Numero di campi dati per classe	17
3.7.3.4	Livello d'annidamento	17
3.7.3.5	Grado di accoppiamento e grado di instabilità	17
3.7.3.6	Chiamate innestate di metodi	18
3.7.3.7	Copertura del codice	18
3.7.3.8	Numero di linee per metodo	18
3.7.3.9	Riepilogo	18
4	Gestione amministrativa della revisione	20
4.1	Comunicazione e risoluzione delle anomalie	20
A	Resoconto dell'attività di verifica	21
A.1	Periodo di Analisi e Management	21
A.1.1	Processi	21
A.1.2	Documenti	21
A.2	Periodo di Analisi di Dettaglio	22
A.2.1	Processi	22

A.2.2	Documenti	22
A.3	Periodo di Progettazione Architettuale	23
A.3.1	Processi	23
A.3.2	Documenti	23
A.4	Periodo di Progettazione di Dettaglio e Codifica	24
A.4.1	Processi	24
A.4.2	Documenti	24
A.5	Periodo di Codifica dei Requisiti Desiderabili e Opzionali	25
A.5.1	Processi	25
A.5.2	Documenti	26
A.6	Periodo di Validazione e Collaudo	26
A.6.1	Processi	26
A.6.2	Documenti	27
A.7	Soddisfacimento metriche	27
A.7.0.1	Complessità ciclomatica	27
A.7.0.2	Grado d'instabilità	28
A.7.0.3	Classi	28
A.7.0.3.1	Client	28
A.7.0.3.2	Server	32
A.7.0.3.3	Client	33
A.7.0.3.4	Server	45
B	Specifica dei test	46
B.1	Descrizione dei test	46
B.2	Modello a V	46
B.3	Test di validazione	46
B.4	Test di sistema	65
B.4.1	Descrizione dei test di sistema	65
B.5	Test di integrazione	66
B.6	Test di unità	68
B.7	Tracciamento test di unità - classe	72
C	Esito delle revisioni	77
C.1	Revisione dei Requisiti	77
C.2	Revisione di Progettazione	77
C.3	Revisione di Qualifica	78

Elenco delle tabelle

1	Riepilogo delle metriche e dei $range_g$ di accettazione e ottimali	19
2	Esiti verifica sui processi - Periodo di Analisi e Management	21
3	Esiti verifica documenti - Periodo di Analisi e Management	22
4	Esiti verifica sui processi - Periodo di Analisi di Dettaglio	22
5	Esiti verifica documenti - Periodo di Analisi di Dettaglio	23
6	Esiti verifica sui processi - Periodo di Progettazione Architettuale	23
7	Esiti verifica documenti - Periodo di Progettazione Architettuale	24
8	Esiti verifica sui processi - Periodo di Progettazione di Dettaglio e Codifica	24
9	Esiti verifica documenti - Periodo di Progettazione di Dettaglio e Codifica	25
10	Esiti verifica sui processi - Periodo di Codifica dei Requisiti Desiderabili e Opzionali	25
11	Esiti verifica documenti - Periodo di Codifica dei Requisiti Desiderabili e Opzionali	26
12	Esiti verifica sui processi - Periodo di Validazione e Collaudo	26
13	Esiti verifica documenti - Periodo di Validazione e Collaudo	27
14	Tabella copertura del codice	27
20	Riepilogo test di validazione	65
21	Riepilogo test di sistema	66
22	Tabella dei test d'integrazione	67
23	Tabella dei test di Unità	72
24	Tabella di tracciamento test di unità - classe	76

Elenco delle figure

1 Modello a V 46

1 Introduzione

1.1 Scopo del documento

Questo documento fissa gli obbiettivi decisi dal $team_g$ per garantire una buona qualità del $prodotto_g$ e dei processi attuati durante l'intera durata del $progetto_g$. Per assicurarne il rispetto verranno svolte costantemente delle attività di verifica dei processi. In questo modo verrà garantita la qualità del $prodotto_g$ e si minimizzano le risorse impiegate.

1.2 Scopo del $prodotto_g$

Il prodotto finale consisterà di un'applicazione mobile che, interagendo con dei beacons sparsi nell'area designata, guiderà l'utente attraverso un percorso. L'utente potrà completare il percorso superando tutte le prove che gli si presenteranno nelle diverse tappe. Le prove potranno essere degli indovinelli o dei semplici giochi inerenti all'area in cui si svolge il percorso.

1.3 Glossario

Al fine di evitare ogni ambiguità nel linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, gli acronimi e le abbreviazioni che necessitano di definizione sono riportati nel documento "*Glossario v3.0.0*".

Inoltre ogni occorrenza di un vocabolo presente nel Glossario sarà posta in corsivo e seguita da una 'g' minuscola a pedice (p.es. $Glossario_g$).

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** "*Norme di Progetto v4.0.0*";
- **Capitolato d'appalto C2:** CLIPS: Communication & Localisation with Indoor Positioning Systems, reperibile all'indirizzo <http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C2.pdf>.

1.4.2 Informativi

- **Analisi dei Requisiti:** "*Analisi dei Requisiti v4.0.0*";
- **Piano di $progetto_g$:** "*Piano di Progetto v4.0.0*";
- **Slide dell'insegnamento Ingegneria del Software modulo A:** <http://www.math.unipd.it/~tullio/IS-1/2015/>;
- **Capacity Maturity Model (CMM_g):** http://en.wikipedia.org/wiki/Capability_Maturity_Model;
- **Capacity Maturity Model Integration (CMMI):** http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration;
- **ISO_g 9001:** http://en.wikipedia.org/wiki/ISO_9001;
- **ISO_g /IEC 9126:2001:** http://en.wikipedia.org/wiki/ISO/IEC_9126;
- **ISO_g /IEC 15504:** http://en.wikipedia.org/wiki/ISO/IEC_15504;
- **Indice $Gulpease_g$:** http://it.wikipedia.org/wiki/Indice_Gulpease.

2 Definizione obiettivi di qualità

2.1 Qualità di prodotto

Basandosi sullo standard [ISO_g/IEC 9126] il *team_g* si impegna a garantire al *prodotto_g* CLIPS le seguenti qualità:

2.1.1 Funzionalità

Il *prodotto_g* deve garantire tutti i requisiti stabiliti nel documento “*Analisi dei Requisiti v4.0.0*” e implementarli nel modo più completo ed economico possibile.

- **Misura:** l’unità di misura adottata sarà la quantità di requisiti presenti e funzionanti nel *prodotto_g*.
- **Metrica:** la sufficienza è stabilita nel soddisfacimento dei requisiti obbligatori.
- **Strumenti:** ogni requisito dovrà superare tutti i test previsti in modo da garantire il loro funzionamento. Per avere informazioni dettagliate sugli strumenti si veda il documento “*Norme di Progetto v4.0.0*”.

2.1.2 Affidabilità

Il *prodotto_g* deve essere il più robusto possibile e facilmente ripristinabile in caso di errori.

- **Misura:** l’unità di misura adottata sarà il numero di esecuzioni che hanno successo.
- **Metrica:** le esecuzioni dovranno coinvolgere tutte le parti possibili del *prodotto_g* ed esaminare il maggior numero possibile di casi. Non si può definire una soglia di sufficienza perché è impossibile determinare ogni situazione d’utilizzo possibile.
- **Strumenti:** Per avere informazioni dettagliate sugli strumenti si veda il documento “*Norme di Progetto v4.0.0*”.

2.2 Usabilità

Il *prodotto_g* deve essere di facile utilizzo per la classe di utenti designata. Inoltre deve soddisfare ogni necessità dell’utente.

- **Misura:** verrà usata come unità di misura la valutazione soggettiva del *prodotto_g*. Questo perché non esiste uno strumento adatto ad eseguire una misurazione oggettiva dell’usabilità.
- **Metrica:** purtroppo non esiste una metrica adeguata che possa determinare una soglia di sufficienza. Il *team_g* si impegna comunque a fornire la miglior qualità d’uso possibile. Per ottenere un risultato più soddisfacente verranno consultate delle persone esterne al gruppo per verificare l’usabilità del *prodotto_g*. Per garantire una misura affidabile (ma comunque non precisa) verranno consultati almeno 15 tester. Tra questi ci saranno almeno 5 persone con una buona confidenza con la tecnologia mobile, 5 con una media confidenza e 5 con una confidenza basilare, cioè persone che possiedono uno *smartphone_g* ma in genere lo usano solo per chiamate, messaggi e email.
- **Strumenti:** si vedano le “*Norme di Progetto v4.0.0*”.

2.3 Efficienza

Il *prodotto_g* deve fornire tutte le funzionalità nel minore tempo possibile e minimizzando l'utilizzo di risorse.

- **Misura:** il tempo di latenza per ottenere una risposta in ogni pagina del *prodotto_g* e quello per ottenere una risposta dal server.
- **Metrica:** la sufficienza è raggiunta nel primo caso con un tempo di latenza inferiore a 0.5 secondi; nel secondo invece è raggiunta con un tempo di latenza minore di 5 secondi, ponendo che non ci siano problemi di connessione.
- **Strumenti:** si vedano le “*Norme di Progetto v4.0.0*”.

2.4 Manutenibilità

Il *prodotto_g* dev'essere comprensibile ed estensibile in modo facile e verificabile.

- **Misura:** l'unità di misura utilizzata saranno le metriche sul codice stabilite nella sezione [Metriche per il codice](#).
- **Metrica:** il *prodotto_g* deve raggiungere la sufficienza in tutte le metriche descritte nella sezione [Metriche per il codice](#).
- **Strumenti:** si vedano le “*Norme di Progetto v4.0.0*”.

2.5 Portabilità

Il *prodotto_g* deve essere il più portabile possibile. Il gruppo Beacon Strips dovrà scegliere un sistema operativo su cui creare l'app tra *Android_g* e *iOS_g*, mentre il supporto per l'altro sarà opzionale. Il *front end_g* dev'essere utilizzabile da più dispositivi possibili. Il *back end_g* deve poter girare su ogni versione dei sistemi operativi supportati che permette la lettura affidabile dei *beacon_g*, ovvero dalla 5.0 in poi per *Android_g* e opionalmente dalla 9.0 in poi per *iOS_g*.

- **Misura:** il *back end_g* dev'essere affidabile per ogni versione di Android ed eventualmente di iOS che supporta la lettura dei *beacon_g*. Ogni dispositivo con una versione dei sistemi operativi supportati uguale o maggiore deve avere un *front end_g* usabile, a prescindere dalle specifiche *hardware_g* o dalla risoluzione dello schermo.
- **Metrica:** il *prodotto_g* dovrà raggiungere la sufficienza in tutte le metriche della sezione [Metriche per il codice](#). Il *back end_g* dovrà raggiungere la sufficienza in affidabilità ed efficienza in ogni dispositivo testato. Il *front end_g* dovrà raggiungere la sufficienza in usabilità in ogni dispositivo testato.
- **Strumenti:** si vedano le “*Norme di Progetto v4.0.0*”.

2.6 Altre qualità

Saranno inoltre garantite le seguenti caratteristiche:

- **incapsulamento:** per aumentare la manutenibilità e il riuso di codice verrà applicata la tecnica dell'incapsulamento. Questo implica che dove sarà possibile verrà favorito l'uso delle interfacce.
- **coesione:** per rendere il *prodotto_g* più manutenibile, più semplice e con un indice di dipendenze minore verrà usata la tecnica della coesione. Questo significa che le funzionalità con il medesimo scopo risiederanno nello stesso componente.

3 Visione generale della strategia

3.1 Procedure di controllo di qualità di processo

Per garantire la qualità dei processi e quindi un loro miglioramento continuo verrà usato il principio *PDCA_g*. Di conseguenza migliorerà la qualità del *prodotto_g*.

Per avere il controllo dei processi, e di conseguenza della qualità, è necessario che:

- i processi siano pianificati dettagliatamente;
- vi sia un controllo sul lavoro di ogni membro del *team_g*;
- nella pianificazione siano ripartite chiaramente le risorse.

L'attuazione di questi punti è approfondita nel “*Piano di Progetto v4.0.0*”.

Analizzando la qualità del *prodotto_g* si controlla anche la qualità dei processi. Un *prodotto_g* scadente indica che i processi sono migliorabili.

Inoltre si possono usare delle metriche per quantificare la qualità dei processi, tali metriche sono descritte nella sezione [Misure e metriche](#).

3.2 Procedure di controllo di qualità di *prodotto_g*

Il controllo di qualità del *prodotto_g* verrà garantito da:

- **quality assurance:** è l'insieme di attività realizzate per raggiungere gli obiettivi di qualità. Queste attività prevedono l'attuazione di tecniche di analisi statica e dinamica, descritte nel documento “*Norme di Progetto v4.0.0*”.
- **verifica:** è un processo che determina se l'output di una fase è consistente, corretto e completo. Per tutta la durata del *progetto_g* verranno svolte attività di verifica, i cui risultati sono e saranno riportati nell'Appendice.
- **validazione:** è il processo di conferma oggettiva del soddisfacimento dei requisiti, attuato per ogni fase del *progetto_g*.

3.3 Organizzazione

L'organizzazione della strategia di verifica si basa sull'attuazione di verifiche per ogni processo completato. Queste verifiche controllano sia la qualità del processo stesso sia la qualità del *prodotto_g* ottenuto. Grazie al diario delle modifiche sarà possibile eseguire una verifica solo sui cambiamenti effettuati.

A causa della diversa natura dei risultati ottenuti da ogni fase del processo ognuno di essi richiederà l'attuazione di specifiche procedure di verifica. Il *team_g* ha deciso di adottare un ciclo di vita *incrementale_g* per lo sviluppo del *progetto_g*. Di conseguenza il processo di verifica adottato per ogni fase del *progetto_g* opererà nel modo seguente:

- **Periodo di Analisi e Management e Periodo di Consolidamento dei Requisiti:** in questa fase verranno redatti i documenti che riporteranno i requisiti individuati, le strategie e le norme adottate.
 - Verrà controllata la correttezza ortografica con il correttore automatico di *TeXstudio_g*.
 - Verrà controllata la correttezza lessicale con un'attenta ed accurata rilettura.
 - Verrà controllata la correttezza dei contenuti rispetto alle aspettative del documento attraverso una rilettura accurata.
 - Verrà verificato che ogni requisito abbia una corrispondenza in un caso d'uso; per farlo si controlleranno le apposite tabelle di tracciamento, con l'ausilio di *Trender_g*.

- Ogni documento dovrà rispettare le “*Norme di Progetto v4.0.0*”; per verificarlo verranno adoperati gli strumenti più appropriati.
- Verrà verificato che sia presente una didascalia per ogni rappresentazione grafica e il contenuto di ogni figura e tabella.
- **Periodo di Progettazione Architettuale:** verrà garantito che ogni requisito possa essere rintracciabile attraverso il processo di verifica. Ogni requisito sarà rintracciabile nei componenti individuati in questa fase, si veda la [Definizione degli obiettivi di qualità](#) per ulteriori approfondimenti.
- **Periodo di Progettazione Architettuale, Periodo di Progettazione di Dettaglio e Codifica e Periodo di Codifica dei Requisiti desiderabili e opzionali:** i Programatori svolgeranno le attività di codifica e di esecuzione dei test di unità per la verifica del codice. Queste attività avverranno nel modo più automatizzato possibile, rispettando anche i vincoli statici. I Verificatori controlleranno parallelamente la presenza di eventuali anomalie, definite nella sezione [Comunicazione e risoluzione delle anomalie](#).
- **Periodo di Validazione e Collaudo:** alla Revisione di Accettazione (RA) il gruppo Beacon Strips garantisce il funzionamento corretto del *prodotto_g* realizzato. Le eventuali modifiche per eliminare le possibili diversità rispetto al *prodotto_g* atteso saranno a carico del *proponente_g*.

In ogni documento viene inoltre incluso il diario delle modifiche, in modo da mantenere uno storico delle attività svolte e delle relative responsabilità.

3.4 Pianificazione strategica e temporale

Per impedire una rapida diffusione degli errori è necessario che la verifica della documentazione sia sistematica ed organizzata. In questo modo inoltre l'individuazione e la correzione degli errori avverrà il prima possibile.

Nel “*Piano di Progetto v4.0.0*” verranno pianificate le attività svolte per migliorare la qualità dei processi, le quali stabiliranno delle nuove norme di *progetto_g*.

Per ridurre la possibilità di commettere errori e/o imprecisioni di natura tecnica/concettuale ogni attività di redazione o di codifica dovrà essere preceduta da uno studio preliminare. In questo modo viene alleggerita l'attività di verifica perché richiederà a posteriori meno interventi correttivi.

Il *team_g* ha come obiettivo il rispetto delle scadenze fissate nel “*Piano di Progetto v4.0.0*”, riportate di seguito:

- revisioni formali:
 - Revisione dei Requisiti: 2016-04-18.
 - Revisione di Accettazione: 2016-09-12.
- revisioni di progresso:
 - Revisione di Progettazione: 2016-06-17.
 - Revisione di Qualifica: 2016-08-24.

3.5 Responsabilità

La responsabilità dell'assegnazione degli incarichi è a carico del Responsabile. L'Amministratore avrà come responsabilità l'adeguamento dell'ambiente di lavoro per lo svolgimento di tutti i compiti necessari alla realizzazione del *progetto_g*. Ogni componente del *team_g* è responsabile del proprio materiale *prodotto_g*.

3.6 Risorse

3.6.1 Necessarie

Per la realizzazione del $prodotto_g$ sono necessarie risorse sia tecnologiche che umane:

- **risorse umane:** vengono descritte dettagliatamente nel “*Piano di Progetto v4.0.0*”.
 - Amministratore;
 - Responsabile;
 - Analista;
 - Progettista;
 - Programmatore;
 - Verificatore.
- **risorse $software_g$:** sono necessari strumenti $software_g$ utili:
 - alla stesura di documentazione in $LaTeX_g$;
 - alla creazione di diagrammi in UML_g ;
 - allo sviluppo nei linguaggi di programmazione scelti;
 - a semplificare ed automatizzare la verifica;
 - all’analisi statica del codice;
 - alla gestione dei test sul codice.
- **risorse $hardware_g$:** sono necessari computer con tutti gli strumenti $software_g$ descritti nelle “*Norme di Progetto v4.0.0*”. È necessario avere a disposizione uno o più luoghi dove poter effettuare le riunioni interne del gruppo Beacon Strips. Sono necessari i $beacon_g$ in un numero sufficiente per i test del $progetto_g$. È necessario avere un server su cui installare il database necessario per il funzionamento del $prodotto_g$. Sono necessari degli $smartphone_g$ adatti ad eseguire i test del $prodotto_g$.

3.6.2 Disponibili

Ogni membro di Beacon Strips ha a disposizione almeno un computer personale dotato di tutti gli strumenti necessari con cui poter svolgere tutti i propri compiti.

A scopo di test, di supporto degli strumenti scelti per l’ambiente di sviluppo, e per il funzionamento dell’app vengono messi a disposizione uno spazio web e un server privato da Miriade SpA.

Per lo svolgimento delle riunioni interne il $team_g$ usa le aule del Dipartimento di Matematica dell’Università degli Studi di Padova; inoltre usa *Google Hangouts_g* come strumento di discussione tramite videochiamate.

3.7 Misure e metriche

Per essere utile ed informativo il processo di verifica dev’essere quantificabile. Vanno quindi stabilite a priori delle metriche, sulle quali saranno basate le misure rilevate dal processo di verifica. Nel caso in cui vi fossero delle metriche approssimate ed incerte, esse miglioreranno in modo *incrementale_g*. Questo grazie al ciclo di vita adottato descritto nel “*Piano di Progetto v4.0.0*”. Possono esserci due tipi di $range_g$:

- **$range_g$ di accettazione:** insieme di valori richiesti affinché il $prodotto_g$ sia accettato.

- **range_g ottimale:** insieme di valori entro i quali dovrebbe collocarsi la misurazione. Esso non è vincolante, ma fortemente consigliato. Scostamenti da tali valori necessitano una verifica approfondita.

3.7.1 Metriche per i processi

3.7.1.1 Modello CMM_g

Il modello CMM_g permette di valutare la maturità e la qualità dei processi. La valutazione viene effettuata tramite una scala di cinque livelli:

1. **Initial:** È il livello più basso. Il SW prodotto è caratterizzato dall'essere costruito seguendo strategie occasionali e caotiche. Solo alcuni processi produttivi sono ben definiti ed i successi dipendono solo da sforzi individuali.
2. **Repeatable:** I processi sono sufficientemente documentati e quindi la loro ripetizione risulta attuabile;
3. **Defined:** I processi sono stabiliti secondo uno standard aziendale e i risultati sono quelli attesi;
4. **Managed:** I processi possono essere controllati grazie alle metriche stabilite, questo consente di identificare possibili miglioramenti ai processi;
5. **Optimizing:** L'efficacia e l'efficienza dei processi sono continuamente migliorati.

Parametri utilizzati:

- **Range accettazione:** Defined - Managed;
- **Range ottimale:** Managed - Optimizing;

3.7.1.2 Schedule Variance

L'avanzamento dei processi verrà valutato tramite la schedule variance. Questo indice di efficacia mette in relazione il lavoro pianificato con quello svolto.

Se l'indice è maggiore di zero allora si è svolto più lavoro rispetto a quello pianificato, viceversa se è negativo. L'utilizzo di periodi di $slack_g$ aumenta la possibilità che la schedule variance assuma valori positivi. **Parametri utilizzati:**

- **Range accettazione:** $\geq -(preventivo\ fase * 10\%)$;
- **Range ottimale:** $\geq -(preventivo\ fase * 5\%)$.

3.7.1.3 Budget Variance

L'avanzamento dei processi verrà valutato anche tramite la budget variance. Questo indice di efficienza mette in relazione il preventivo pianificato con il consuntivo.

Se l'indice è maggiore di zero allora si è riusciti ad essere più efficienti di quello pianificato, viceversa se negativo. **Parametri utilizzati:**

- **Range accettazione:** $\geq (preventivo\ fase * 10\%)$;
- **Range ottimale:** $\geq (preventivo\ fase * 5\%)$.

3.7.2 Metriche per i documenti

3.7.2.1 Indice *Gulpease_g*

L'indice *Gulpease_g* è un indice di leggibilità del testo creato appositamente per valutare la lingua italiana. Dato che non valuta la lunghezza delle parole mediante il numero di sillabe tale indice semplifica il calcolo rispetto ad altri indici di leggibilità. Esso infatti è basato sul numero di caratteri contenuti in una parola e ad altri fattori come il numero di parole e di frasi. Come tutti gli indici di leggibilità esso permette di indicare la complessità di un documento. Il calcolo da effettuare è il seguente:

$$89 + \frac{300(\text{numero delle frasi}) - 10(\text{numero delle lettere})}{\text{numero delle parole}}$$

I risultati sono compresi tra 0 e 100, dove 100 è il valore di leggibilità più alto e 0 quello più basso. Sono state stabilite delle soglie per rapportare il livello di istruzione di un individuo con i vari gradi dell'indice:

- **inferiore a 80:** documento difficile da leggere per chi ha la licenza elementare;
- **inferiore a 60:** documento difficile da leggere per chi ha licenza media;
- **inferiore a 40:** documento difficile da leggere per chi ha la licenza superiore.

In realtà tale indice non certifica se il testo sia comprensibile o meno. Per lo scopo dei documenti e per la formalità richiesti da essi verranno usati spesso termini tecnici che non si possono sostituire. Di conseguenza un documento potrebbe avere un ottimo indice di *Gulpease_g* ma essere difficile da comprendere per i termini usati. Anche spezzare una frase può migliorare l'indice, ma interromperebbe il ragionamento fatto in quella frase. Infine usare frasi troppo dirette potrebbe essere poco professionale ai fini del documento. Per queste ragioni i documenti verranno valutati da un essere umano, così potrà stabilire se il testo è semplificabile. I limiti imposti da tale indice saranno sufficientemente rilassati per accettare frasi un po' più articolate.

Parametri utilizzati:

- **Range accettazione:** 40 - 100;
- **Range ottimale:** 50 - 100.

3.7.3 Metriche per il codice

3.7.3.1 Numero di parametri

Indica il numero di parametri formali in input di un metodo. Se il numero di parametri è elevato, lo *stack_g* del programma può essere riempito rapidamente in caso di chiamate multiple innestate. I costruttori potranno superare questo limite se il numero elevato di parametri potrà facilitare la verifica tramite test e rendere più robusto agli errori l'oggetto definito.

Parametri utilizzati:

- **Range accettazione:** 0 - 8;
- **Range ottimale:** 0 - 5.

3.7.3.2 Complessità ciclomatica

Indica il numero di cammini linearmente indipendenti che attraversano il grafo di flusso di controllo del metodo. In tale grafo i nodi rappresentano unità atomiche di istruzioni. Gli archi indicano che le istruzioni collegate dai nodi collegati possono essere eseguite consecutivamente.

Un alto valore di complessità si può ridurre con la suddivisione in più metodi. È accettata anche

una misurazione più lasca se questo dovesse influire in modo positivo sulla velocità di esecuzione. Oltretutto un valore alto potrebbe essere causato da strutture che in realtà aiutano ad ordinare il codice, come gli switch di condizioni.

Parametri utilizzati:

- **Range accettazione:** 0 - 10;
- **Range ottimale:** 0 - 6.

3.7.3.3 Numero di campi dati per classe

Un elevato numero di attributi interni rende la classe troppo poco specializzata, ed è indice di cattiva progettazione. Dato che la classe ricoprirà più ruoli, rende anche più difficile il mantenimento del codice.

La riduzione del numero dei campi dati si può ottenere con l'incapsulamento in nuove classi.

Parametri utilizzati:

- **Range accettazione:** 0 - 16;
- **Range ottimale:** 0 - 10.

3.7.3.4 Livello d'annidamento

Indica quante volte le strutture di controllo sono state inserite l'una all'interno dell'altra. Un alto valore può portare a difficoltà nella verifica e nell'astrazione del codice.

Parametri utilizzati:

- **Range accettazione:** 0 - 6;
- **Range ottimale:** 0 - 4.

3.7.3.5 Grado di accoppiamento e grado di instabilità

Il grado di accoppiamento è derivato da due singoli indici.

- **Accoppiamento afferente:** numero di classi esterne al *package_g* che dipendono da classi al suo interno. Se il numero è basso il *package_g* non fornirà molte funzionalità e sarà poco utile. Se invece è alto troppe classi saranno dipendenti da tale *package_g*, rischiando di dover effettuare troppi cambiamenti ad ogni sua modifica.
- **Accoppiamento efferente:** numero di classi interne al *package_g* dipendenti da classi esterne. Un alto numero può indicare una scarsa progettazione.

Il grado di instabilità invece misura l'instabilità delle componenti del sistema utilizzando i due indici precedenti.

La stabilità di una componente indica la possibilità di effettuare modifiche a quella componente senza influenzarne altre all'interno dell'applicazione. Questo indice viene calcolato attraverso la formula:

$$I = \frac{Ce}{Ca + Ce}$$

dove Ce rappresenta l'accoppiamento efferente e Ca quello afferente.

Parametri utilizzati:

- **Range accettazione:** 0 - 0.8;
- **Range ottimale:** 0.3 - 0.7.

3.7.3.6 Chiamate innestate di metodi

Un grande numero di chiamate innestate di metodi può portare alla saturazione dello $stack_g$, soprattutto in caso di un grande numero di parametri, quindi è necessario limitarne il numero.

Parametri utilizzati:

- **Range accettazione:** 0 - 8;
- **Range ottimale:** 0 - 5.

3.7.3.7 Copertura del codice

Rappresenta la percentuale di istruzioni eseguite durante i test. Maggiore è questo valore più significativi saranno i test eseguiti, quindi saranno minori le probabilità di rilevare errori residui al termine dei test.

Si può ridurre l'indice tramite l'utilizzo di metodi semplici che non richiedono test, come i metodi getter e setter.

Parametri utilizzati:

- **Range accettazione:** 80% – 100%;
- **Range ottimale:** 90% – 100%.

3.7.3.8 Numero di linee per metodo

Indica il numero di $statement_g$ che compongono un metodo. Maggiore è la quantità, più lungo è il metodo, più difficile risulta comprenderne il funzionamento.

Se un metodo risulta troppo lungo allora può servire dividerlo in più sottofunzioni. Può anche essere indice di cattiva progettazione della classe, per cui risulta essere necessaria anche una riprogettazione per intero.

Parametri utilizzati:

- **Range accettazione:** al massimo 50;
- **Range ottimale:** al massimo 40.

3.7.3.9 Riepilogo

Metriche	Range accettazione	Range ottimale
Metriche per i processi		
Modello CMM_g	Defined - Managed	Managed - Optimizing
Schedule Variance	$\geq -10\%$	$\geq -5\%$
Budget Variance	$\geq -10\%$	$\geq -5\%$
Metriche per i documenti		
Indice $Gulpease_g$	40 - 100	50 - 100
Metriche per il codice		
Numero di parametri	0 - 8	0 - 5
Complessità ciclomatica	0 - 10	0 - 6
Numero di campi dati per classe	0 - 16	0 - 10

Metriche	Range accettazione	Range ottimale
Livello d'annidamento	0 - 6	0 - 4
Grado di instabilità	0 - 0.8	0.3 - 0.7
Chiamate innestate di metodi	0 - 8	0 - 5
Copertura del codice	80% - 100%	90% - 100%
Numero di linee per metodo	≤ 50	≤ 40

Tabella 1: Riepilogo delle metriche e dei $range_g$ di accettazione e ottimali

4 Gestione amministrativa della revisione

4.1 Comunicazione e risoluzione delle anomalie

Un'anomalia corrisponde a:

- un errore ortografico;
- la violazione delle norme tipografiche del documento;
- l'uscita dal $range_g$ di accettazione degli indici di misurazione, descritti nella sezione [Misure e metriche](#);
- un'incongruenza del $prodotto_g$ rispetto a determinate funzionalità. Tali funzionalità sono indicate nel documento “*Analisi dei Requisiti v4.0.0*”;
- un'incongruenza del codice con il design del $prodotto_g$.

Nel caso in cui un Verificatore individui un'anomalia, dovrà aprire un $ticket_g$ seguendo la procedura indicata nelle “*Norme di Progetto v4.0.0*”.

A Resoconto dell'attività di verifica

A.1 Periodo di Analisi e Management

A.1.1 Processi

Sono riportati ora i valori di Schedule Variance e Budget Variance per le attività del Periodo di Analisi e Management.

Attività	Schedule Variance	Budget Variance
"Analisi dei Requisiti v4.0.0"	€ -10	€ +15
"Glossario v3.0.0"	€ 0	€ 0
"Norme di Progetto v4.0.0"	€ -5	€ +10
"Piano di Progetto v4.0.0"	€ 0	€ +20
"Piano di Qualifica v4.0.0"	€ +5	€ -10
"Studio di Fattibilità v2.0.0"	€ -10	€ 0

Tabella 2: Esiti verifica sui processi - Periodo di Analisi e Management

In totale sono stati registrati:

- **Schedule Variance:** € -20;
- **Budget Variance:** € +35.

Dai valori ottenuti si nota subito che in alcuni casi è stata prevista qualche ora di attività in più rispetto al necessario. Questo ha portato ad avere una Budget Variance positiva, mentre la causa di questo fatto è da ricercare nell'inesperienza del gruppo, che ha portato ad una valutazione leggermente errata del carico di lavoro necessario. Sempre a causa della poca esperienza del gruppo la Schedule Variance è risultata negativa, perché alcune attività si sono concluse leggermente in ritardo rispetto alle aspettative. La causa di questo fatto è da ricercare nell'organizzazione da parte dei membri del gruppo dei propri impegni, da cui sono derivati leggeri ritardi che si sono sommati. Il risultato comunque rientra nel limite ottimale, che sarebbe di € -144.

A.1.2 Documenti

Sono riportati qui i valori dell'indice Gulpease per ogni documento presente durante l'attività di analisi nel Periodo di Analisi e Management. Un documento è considerato valido soltanto se rispetta le metriche descritte secondo la sezione [Indice Gulpease](#).

Documento	Valore	Esito
"Analisi dei Requisiti v4.0.0"	66	Superato
"Glossario v3.0.0"	53	Superato
"Norme di Progetto v4.0.0"	56	Superato
"Piano di Progetto v4.0.0"	53	Superato
"Piano di Qualifica v4.0.0"	60	Superato
"Studio di Fattibilità v2.0.0"	61	Superato

Documento	Valore	Esito
-----------	--------	-------

Tabella 3: Esiti verifica documenti - Periodo di Analisi e Management

A.2 Periodo di Analisi di Dettaglio

A.2.1 Processi

Sono riportati ora i valori di Schedule Variance e Budget Variance per le attività del Periodo di Analisi di Dettaglio.

Attività	Schedule Variance	Budget Variance
"Analisi dei Requisiti v4.0.0"	€ -20	€ -25
"Glossario v3.0.0"	€ 0	€ 0
"Norme di Progetto v4.0.0"	€ +5	€ +5
"Piano di Progetto v4.0.0"	€ -10	€ -10
"Piano di Qualifica v4.0.0"	€ 0	€ -5
"Studio di Fattibilità v2.0.0"	€ 0	€ 0

Tabella 4: Esiti verifica sui processi - Periodo di Analisi di Dettaglio

In totale sono stati registrati:

- **Schedule Variance:** € -25;
- **Budget Variance:** € -35.

Dai valori ottenuti si nota subito che il lavoro richiesto per eseguire le attività è stato maggiore di quello pianificato. Questo perché le modifiche necessarie si sono rivelate essere più del previsto, soprattutto per quanto riguarda l'Analisi dei Requisiti. Di conseguenza la Budget Variance e la Schedule Variance sono risultate negative, ma comunque al di sotto del limite ottimale di € -65.

A.2.2 Documenti

Sono riportati qui i valori dell'indice Gulpease per ogni documento presente durante l'attività di analisi nel Periodo di Analisi di Dettaglio. Un documento è considerato valido soltanto se rispetta le metriche descritte secondo la sezione [Indice Gulpease](#).

Documento	Valore	Esito
"Analisi dei Requisiti v4.0.0"	85	Superato
"Glossario v3.0.0"	54	Superato
"Norme di Progetto v4.0.0"	51	Superato
"Piano di Progetto v4.0.0"	62	Superato
"Piano di Qualifica v4.0.0"	61	Superato
"Studio di Fattibilità v2.0.0"	61	Superato

Documento	Valore	Esito
-----------	--------	-------

Tabella 5: Esiti verifica documenti - Periodo di Analisi di Dettaglio

A.3 Periodo di Progettazione Architettuale

A.3.1 Processi

Sono riportati ora i valori di Schedule Variance e Budget Variance per le attività del Periodo di Progettazione Architettuale.

Attività	Schedule Variance	Budget Variance
"Analisi dei Requisiti v4.0.0"	€ +20	€ +20
"Glossario v3.0.0"	€ +5	€ 0
"Norme di Progetto v4.0.0"	€ +10	€ +15
"Piano di Progetto v4.0.0"	€ -5	€ 0
"Piano di Qualifica v4.0.0"	€ -5	€ 0
"Specifica Tecnica v3.0.0"	€ -20	€ -50

Tabella 6: Esiti verifica sui processi - Periodo di Progettazione Architettuale

In totale sono stati registrati:

- **Schedule Variance:** € +5;
- **Budget Variance:** € -15.

Dai valori ottenuti si nota subito che le attività si sono concluse al massimo leggermente dopo i tempi previsti, ad eccezione della Specifica Tecnica che ha subito notevoli ritardi.

Questo ha portato ad avere una Schedule Variance leggermente positiva perché il tempo risparmiato ha compensato quello perso per la Specifica Tecnica, in particolare sono risultate importanti le ore dedicate nella fase precedente perché ha permesso di avere dei documenti praticamente già terminati.

Al contrario la budget Variance è risultata negativa ma comunque al di sotto del limite ottimale di € -204. Questo perché le ore di lavoro previste per tutti i documenti sono risultate sufficienti o addirittura eccessive, ma la Specifica Tecnica ha richiesto molte più ore del previsto.

I dati negativi relativi a questo documento sono da ricercare sempre nell'inesperienza del gruppo, che ha portato a sopravvalutare non di poco i tempi e le ore richiesti per portarlo a termine, in particolare per quanto riguarda la progettazione delle classi.

A.3.2 Documenti

Sono riportati qui i valori dell'indice Gulpease per ogni documento presente durante l'attività di analisi nel Periodo di Progettazione Architettuale. Un documento è considerato valido soltanto se rispetta le metriche descritte secondo la sezione [Indice Gulpease](#).

Documento	Valore	Esito
"Analisi dei Requisiti v4.0.0"	84	Superato
"Glossario v3.0.0"	54	Superato

Documento	Valore	Esito
“Norme di Progetto v4.0.0”	71	Superato
“Piano di Progetto v4.0.0”	51	Superato
“Piano di Qualifica v4.0.0”	61	Superato
“Specifica Tecnica v3.0.0”	86	Superato

Tabella 7: Esiti verifica documenti - Periodo di Progettazione Architettuale

A.4 Periodo di Progettazione di Dettaglio e Codifica

A.4.1 Processi

Sono riportati ora i valori di Schedule Variance e Budget Variance per le attività del Periodo di Progettazione di Dettaglio e Codifica.

Attività	Schedule Variance	Budget Variance
“Analisi dei Requisiti v4.0.0”	€ +10	€ +10
“Definizione di Prodotto v2.0.0”	€ +15	€ +20
“Glossario v3.0.0”	€ +5	€ +10
“Manuale Utente v2.0.0”	€ +20	€ +15
“Norme di Progetto v4.0.0”	€ -5	€ 0
“Piano di Progetto v4.0.0”	€ +10	€ +20
“Piano di Qualifica v4.0.0”	€ +5	€ 0
“Specifica Tecnica v3.0.0”	€ -10	€ -15

Tabella 8: Esiti verifica sui processi - Periodo di Progettazione di Dettaglio e Codifica

In totale sono stati registrati:

- **Schedule Variance:** € +15;
- **Budget Variance:** € +10.

Dai valori ottenuti si nota subito che le attività sono state concluse in anticipo rispetto alle previsioni, e con una quantità di lavoro minore di quella prevista. I valori di Schedule Variance e di Budget Variance sono risultati quindi positivi, perché la correzione di alcuni documenti ha richiesto meno tempo del previsto. Questo risultato è da cercarsi anche nel maggior carico di lavoro richiesto dalla parte di codifica, che quindi ha portato il gruppo a sistemare presto i documenti.

A.4.2 Documenti

Sono riportati qui i valori dell'indice Gulpease per ogni documento presente durante l'attività di analisi nel Periodo di Progettazione di Dettaglio e Codifica. Un documento è considerato valido soltanto se rispetta le metriche descritte secondo la sezione [Indice Gulpease](#).

Documento	Valore	Esito
-----------	--------	-------

Documento	Valore	Esito
“Analisi dei Requisiti v4.0.0”	79	Superato
“Definizione di Prodotto v2.0.0”	79	Superato
“Glossario v3.0.0”	48	Superato
“Manuale Utente v2.0.0”	50	Superato
“Norme di Progetto v4.0.0”	70	Superato
“Piano di Progetto v4.0.0”	51	Superato
“Piano di Qualifica v4.0.0”	52	Superato
“Specifica Tecnica v3.0.0”	87	Superato

Tabella 9: Esiti verifica documenti - Periodo di Progettazione di Dettaglio e Codifica

A.5 Periodo di Codifica dei Requisiti Desiderabili e Opzionali

A.5.1 Processi

Sono riportati ora i valori di Schedule Variance e Budget Variance per le attività del Periodo di Codifica dei Requisiti Desiderabili e Opzionali.

Attività	Schedule Variance	Budget Variance
“Analisi dei Requisiti v4.0.0”	€ +10	€ +10
“Definizione di Prodotto v2.0.0”	€ 0	€ 0
“Glossario v3.0.0”	€ 0	€ 0
“Manuale Utente v2.0.0”	€ +5	€ +10
“Norme di Progetto v4.0.0”	€ +10	€ 0
“Piano di Progetto v4.0.0”	€ +10	€ +5
“Piano di Qualifica v4.0.0”	€ +5	€ +5
“Specifica Tecnica v3.0.0”	€ +10	€ +5

Tabella 10: Esiti verifica sui processi - Periodo di Codifica dei Requisiti Desiderabili e Opzionali

In totale sono stati registrati:

- **Schedule Variance:** € +50;
- **Budget Variance:** € +35.

Dai valori ottenuti si nota subito che le attività sono state concluse leggermente in anticipo rispetto alle previsioni, e con una quantità di lavoro un pochino minore di quella prevista. I valori di Schedule Variance e di Budget Variance sono risultati quindi positivi. Questo risultato è da cercarsi nel fatto che l'uscita dell'esito della Revisione di Qualifica era previsto all'incirca per la fine del periodo, come è avvenuto di fatto; di conseguenza non sono state previste molte ore di attività per i documenti, oltre a quelle necessarie per mantenerli aggiornati.

A.5.2 Documenti

Sono riportati qui i valori dell'indice Gulpease per ogni documento presente durante l'attività di analisi nel Periodo di Codifica dei Requisiti Desiderabili e Opzionali. Un documento è considerato valido soltanto se rispetta le metriche descritte secondo la sezione [Indice Gulpease](#).

Documento	Valore	Esito
"Analisi dei Requisiti v4.0.0"	79	Superato
"Definizione di Prodotto v2.0.0"	78	Superato
"Glossario v3.0.0"	48	Superato
"Manuale Utente v2.0.0"	49	Superato
"Norme di Progetto v4.0.0"	70	Superato
"Piano di Progetto v4.0.0"	50	Superato
"Piano di Qualifica v4.0.0"	48	Superato
"Specifica Tecnica v3.0.0"	87	Superato

Tabella 11: Esiti verifica documenti - Periodo di Codifica dei Requisiti Desiderabili e Opzionali

A.6 Periodo di Validazione e Collaudo

A.6.1 Processi

Sono riportati ora i valori di Schedule Variance e Budget Variance per le attività del Periodo di Validazione e Collaudo.

Attività	Schedule Variance	Budget Variance
"Analisi dei Requisiti v4.0.0"	€ +5	€ +5
"Definizione di Prodotto v2.0.0"	€ -30	€ -35
"Glossario v3.0.0"	€ +5	€ +5
"Manuale Utente v2.0.0"	€ -10	€ -10
"Norme di Progetto v4.0.0"	€ +5	€ +10
"Piano di Progetto v4.0.0"	€ +10	€ +15
"Piano di Qualifica v4.0.0"	€ +10	€ +5
"Specifica Tecnica v3.0.0"	€ -5	€ -10

Tabella 12: Esiti verifica sui processi - Periodo di Validazione e Collaudo

In totale sono stati registrati:

- **Schedule Variance:** € -10;
- **Budget Variance:** € -15.

Dai valori ottenuti si nota subito che le attività sono state concluse in leggero ritardo rispetto alle previsioni, e con una quantità di lavoro un po' maggiore di quella prevista. I valori di Schedule Variance e di Budget Variance sono risultati quindi negativi, ma comunque minori del limite ottimale

di € -60. Questo risultato è da cercarsi nel maggior numero di correzioni richieste dall'esito della Revisione di Qualifica rispetto a quelle previste, che di conseguenza hanno richiesto un maggior carico di lavoro, in particolare per quanto riguarda la *DPdoc_g*.

A.6.2 Documenti

Sono riportati qui i valori dell'indice Gulpease per ogni documento presente durante l'attività di analisi nel Periodo di Validazione e Collaudo. Un documento è considerato valido soltanto se rispetta le metriche descritte secondo la sezione [Indice Gulpease](#).

Documento	Valore	Esito
"Analisi dei Requisiti v4.0.0"	79	Superato
"Definizione di Prodotto v2.0.0"	77	Superato
"Glossario v3.0.0"	48	Superato
"Manuale Utente v2.0.0"	49	Superato
"Norme di Progetto v4.0.0"	70	Superato
"Piano di Progetto v4.0.0"	50	Superato
"Piano di Qualifica v4.0.0"	62	Superato
"Specifica Tecnica v3.0.0"	87	Superato

Tabella 13: Esiti verifica documenti - Periodo di Validazione e Collaudo

A.7 Soddisfacimento metriche

A.7.0.1 Complessità ciclomatica

Di seguito viene riportata la tabella riassuntiva della copertura del codice.

Componente	Percentuale	Esito
urlrequest	92%	Superato
datamanager	94 %	Superato
data	92%	Superato
pathprogress	29%	Non Superato
authentication	0%	Non Superato
building	0%	Non Superato
games	0%	Non Superato
savedresults	0%	Non Superato
utility	0%	Non Superato
server	64%	Non Superato

Tabella 14: Tabella copertura del codice

A.7.0.2 Grado d'instabilità

Viene riportata di seguito una tabella riassuntiva che espone gli indici di instabilità per i componenti rilevati durante la progettazione.

Componente	Indice d'instabilità	Esito
CLIPS	0	Superato
CLIPS::client	0	Superato
CLIPS::client::authentication	0	Superato
CLIPS::client::data	0	Superato
CLIPS::client::data::datamanager	0.02	Superato
CLIPS::client::data::urlrequest	0.8	Superato
CLIPS::client::pathprogress	0	Superato
CLIPS::client::viewController	0	Superato
CLIPS::client::viewController::building	0	Superato
CLIPS::client::viewController::games	0	Superato
CLIPS::client::viewController::savedresults	0	Superato
CLIPS::client::viewController::utility	0	Superato
CLIPS::server	0	Superato
CLIPS::server::data	0	Superato
CLIPS::server::urlrequesthandler	0	Superato

A.7.0.3 Classi

Vengono riportate di seguito le tabelle riassuntive che espongono gli esiti delle metriche sulle classi di client e server.

A.7.0.3.1 Client

Classe	Numero attributi	Esito
AccountActivity	2	Superato
AccountView	0	Superato
ConfirmRegistration	0	Superato
ForgotPasswordActivity	0	Superato
ForgotPasswordView	0	Superato
LoginActivity	4	Superato
LoginView	0	Superato
RecoverPassword	0	Superato
RegistrationActivity	8	Superato
RegistrationView	0	Superato

Classe	Numero attributi	Esito
UpdateUserInfoActivity	0	Superato
UpdateUserInfoView	0	Superato
AbstractQuiz	2	Superato
AppInfo	4	Superato
Beacon	4	Superato
BiggerShape	4	Superato
BiggerShapeTest	3	Superato
Building	15	Superato
GameCollection	2	Superato
LeaderBoard	5	Superato
LinearScoringAlgorithm	6	Superato
LoggedUser	3	Superato
MultipleChoiceImageQuiz	1	Superato
MultipleChoiceTest	3	Superato
MultipleChoiceTextQuiz	2	Superato
Path	4	Superato
PathInfo	6	Superato
PathProgress	4	Superato
PathResult	7	Superato
Proof	7	Superato
ProofResult	4	Superato
Proximity	3	Superato
Score	3	Superato
ScoringAlgorithmFactory	0	Superato
Step	4	Superato
Test	0	Superato
TestBuilder	0	Superato
TrueFalseImageQuiz	1	Superato
TrueFalseTest	2	Superato
TrueFalseTextQuiz	1	Superato
Utility	0	Superato
AbstractDataManagerListener	0	Superato
AppInfoDataRequest	0	Superato
BuildingsDataRequest	4	Superato
CheckResult	4	Superato
DataManager	5	Superato

Classe	Numero attributi	Esito
DataRequestMaker	0	Superato
DBHandler	2	Superato
GetRankingDataRequest	1	Superato
GetResultsDataRequest	0	Superato
LoginManager	3	Superato
PathDataRequest	1	Superato
SaveDataManager	2	Superato
SaveResultDataRequest	1	Superato
AbstractUrlRequestListener	0	Superato
AppInfoRequest	0	Superato
BuildingsRequest	0	Superato
ChangeProfileDataRequest	0	Superato
CheckRequest	0	Superato
ForgotPasswordRequest	0	Superato
GetProfileDataRequest	0	Superato
GetRankingRequest	0	Superato
GetResultsRequest	0	Superato
LoginRequest	0	Superato
LogoutRequest	0	Superato
PathRequest	0	Superato
RegistrationRequest	0	Superato
RequestMaker	0	Superato
SaveResultRequest	0	Superato
ServerError	3	Superato
URLDataConstants	2	Superato
URLRequest	7	Superato
BeaconDiscover	1	Superato
BeaconDiscoverDelegate	0	Superato
GPSListener	0	Superato
GPSManager	5	Superato
PathProgressController	4	Superato
PathProgressControllerDelegate	0	Superato
PahtProgressMaker	0	Superato
ProximityManagerPath	4	Superato
RawBeacon	3	Superato
BuildingActivity	12	Superato

Classe	Numero attributi	Esito
BuildingRow	2	Superato
BuildingSearchActivity	7	Superato
BuildingSearchView	0	Superato
BuildingView	0	Superato
PathActivity	3	Superato
PathRow	1	Superato
PathView	0	Superato
SearchCriteriaRow	4	Superato
MultipleChoiceQuizActivity	7	Superato
MultipleChoiceQuizView	0	Superato
ProofActivity	0	Superato
ProofResultActivity	8	Superato
ProofResultRow	0	Superato
ProofResultView	0	Superato
ProofView	0	Superato
QuizActivity	0	Superato
QuizResult View	2	Superato
QuizView	1	Superato
SearchNewStepActivity	8	Superato
SearchNewStepView	0	Superato
TestResultView	0	Superato
TestView	0	Superato
TrueFalseQuizActivity	6	Superato
TrueFalseQuizView	0	Superato
PlayerRow	2	Superato
RankingActivity	1	Superato
RankingView	0	Superato
ResultActivity	4	Superato
ResultRow	5	Superato
ResultView	0	Superato
SavedResultsActivity	0	Superato
SavedResult View	0	Superato
AppInfoActivity	0	Superato
AppInfoView	0	Superato
BuildingAdapter	2	Superato
MenuActivity	6	Superato

Classe	Numero attributi	Esito
MenuView	0	Superato
RankingAdapter	2	Superato
ResultsAdapter	2	Superato
SplashActivity	1	Superato

A.7.0.3.2 Server

Classe	Numero attributi	Esito
AppInfo	0	Superato
AuthToken	0	Superato
Beacon	0	Superato
Building	0	Superato
LeaderBoard	0	Superato
PathResult	0	Superato
Path	0	Superato
ProofResult	0	Superato
Proof	0	Superato
Proximity	0	Superato
ScoreManager	0	Superato
Step	0	Superato
User	0	Superato
AppInfoProvider	0	Superato
BuildingsProvider	0	Superato
DBHandler	1	Superato
EmailChecker	0	Superato
LoginHandler	0	Superato
LogoutHandler	0	Superato
PasswordChecker	1	Superato
PathProvider	0	Superato
PathResultsProvider	0	Superato
RegistrationFieldsValidator	0	Superato
RegistrationHandler	0	Superato
URLRequestHandler	2	Superato
URLRequestRouter	0	Superato
UsernameChecker	0	Superato
UserResultProvider	0	Superato

Metodi Vengono riportate di seguito le tabelle riassuntive che espongono gli esiti delle metriche sui metodi di classe.

Per motivi di spazio le colonne sono state rinominate nel seguente modo:

- **LC**: linee di codice;
- **LA**: livello di annidamento;
- **CI**: numero di chiamate innestate di metodi;
- **NP**: numero di parametri;
- **CC**: complessità ciclomatica.

A.7.0.3.3 Client

::Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::Building::getPathInfo()	3	0	0	0	1	Superato
data::datamanager::AbstractDataManagerListener::onError(ServerError)	0	0	0	1	0	Superato
data::datamanager::AbstractDataManagerListener::onResponse(ModelObject)	0	0	0	1	0	Superato
data::datamanager::AppInfoDataRequest::getRemoteData(AbstractUrlRequestListener)	3	0	0	1	1	Superato
data::datamanager::AppInfoDataRequest::parseFromLocal()	5	0	0	0	1	Superato
data::datamanager::AppInfoDataRequest::parseFromUrlRequest(JSONObject)	9	0	1	1	2	Superato
data::datamanager::AppInfoDataRequest::updateLocalData(AppInfo)	13	0	0	1	1	Superato
data::datamanager::BuildingsDataRequest::getRemoteData(AbstractUrlRequestListener)	3	0	0	1	1	Superato
data::datamanager::BuildingsDataRequest::parseFromLocal()	3	0	0	0	1	Superato
data::datamanager::BuildingsDataRequest::parseFromUrlRequest(JSONObject)	20	2	3	1	4	Superato
data::datamanager::BuildingsDataRequest::updateLocalData(Building[])	3	0	0	1	1	Superato
data::datamanager::DataManager::execute()	1	0	4	0	5	Superato
data::datamanager::DataManager::getRemoteData(AbstractUrlRequestListener)	0	0	0	1	0	Superato
data::datamanager::DataManager::parseFromLocal()	0	0	0	0	0	Superato
data::datamanager::DataManager::parseFromUrlRequest(JSONObject)	0	0	0	1	0	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::datamanager::DataManager::updateLocalData(Data)	0	0	0	1	0	Superato
data::datamanager::DataRequestMaker::getAppInfo(Context, AbstractDataManagerListener <AppInfo>)	3	0	0	2	1	Superato
data::datamanager::DataRequestMaker::getBuildings(Context, double, double, int, boolean, AbstractDataManagerListener <Building[]>)	3	0	0	6	1	Superato
data::datamanager::DataRequestMaker::getPath(Context, int, AbstractDataManagerListener <Path>)	3	0	0	3	1	Superato
data::datamanager::DataRequestMaker::getRanking(Context, int, AbstractDataManagerListener <Score[]>)	3	0	0	3	1	Superato
data::datamanager::DataRequestMaker::getResults(Context, AbstractDataManagerListener <PathResult[]>)	3	0	0	2	1	Superato
data::datamanager::DataRequestMaker::saveResult(Context, PathResult, AbstractDataManagerListener <Boolean>)	3	0	0	3	1	Superato
data::datamanager::DBHandler::createBeaconTable(SQLiteDatabase)	5	0	0	1	1	Superato
data::datamanager::DBHandler::createBuildingTable(SQLiteDatabase)	7	0	0	1	1	Superato
data::datamanager::DBHandler::createPathInfoTable(SQLiteDatabase)	7	0	0	1	1	Superato
data::datamanager::DBHandler::createPathResultTable(SQLiteDatabase)	6	0	0	1	1	Superato
data::datamanager::DBHandler::createPathTable(SQLiteDatabase)	5	0	0	1	1	Superato
data::datamanager::DBHandler::createProofResultTable(SQLiteDatabase)	7	0	0	1	1	Superato
data::datamanager::DBHandler::createProofTable(SQLiteDatabase)	9	0	0	1	1	Superato
data::datamanager::DBHandler::createProximityTable(SQLiteDatabase)	7	0	0	1	1	Superato
data::datamanager::DBHandler::createStepTable(SQLiteDatabase)	8	0	0	1	1	Superato
data::datamanager::DBHandler::DBHandler(Context)	3	0	0	1	1	Superato
data::datamanager::DBHandler::deleteAllBuildings()	6	0	0	0	1	Superato
data::datamanager::DBHandler::deleteAllPathInfos()	5	0	0	0	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::datamanager::DBHandler::deleteAllProofsResults()	5	0	0	0	1	Superato
data::datamanager::DBHandler::deleteAllProximities()	5	0	0	0	1	Superato
data::datamanager::DBHandler::deleteAllSteps()	5	0	0	0	1	Superato
data::datamanager::DBHandler::deleteBuilding(int)	6	0	0	1	1	Superato
data::datamanager::DBHandler::deletePath(int)	7	0	0	1	1	Superato
data::datamanager::DBHandler::deletePathInfo(int)	5	0	0	1	1	Superato
data::datamanager::DBHandler::deletePathInfos(int)	8	1	1	1	2	Superato
data::datamanager::DBHandler::deletePathResult(int)	6	0	0	1	1	Superato
data::datamanager::DBHandler::deletePathResults()	6	0	0	0	1	Superato
data::datamanager::DBHandler::deleteProof(int)	5	0	0	1	1	Superato
data::datamanager::DBHandler::deleteProofResult(int)	5	0	0	1	1	Superato
data::datamanager::DBHandler::deleteProofsResults(int)	5	0	0	1	1	Superato
data::datamanager::DBHandler::deleteProximities(int)	5	0	0	1	1	Superato
data::datamanager::DBHandler::deleteProximity(int)	5	0	0	1	1	Superato
data::datamanager::DBHandler::deleteStep(int)	6	0	0	1	1	Superato
data::datamanager::DBHandler::deleteSteps(int)	8	1	1	1	2	Superato
data::datamanager::DBHandler::getBuildingName(int)	11	1	1	1	2	Superato
data::datamanager::DBHandler::getNearestBuildings(int, boolean, double, double)	10	1	1	4	2	Superato
data::datamanager::DBHandler::getStepID(Step)	9	1	1	1	2	Superato
data::datamanager::DBHandler::onCreate(SQLiteDatabase)	12	0	0	1	1	Superato
data::datamanager::DBHandler::onUpgrade(SQLiteDatabase, int, int)	4	0	0	3	1	Superato
data::datamanager::DBHandler::readBeacon(int)	13	0	1	1	2	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::datamanager::DBHandler::readBuilding(int)	25	0	1	1	2	Superato
data::datamanager::DBHandler::readBuildings()	25	1	1	0	2	Superato
data::datamanager::DBHandler::readPath(int)	14	0	1	1	2	Superato
data::datamanager::DBHandler::readPathInfo(int)	14	1	1	1	2	Superato
data::datamanager::DBHandler::readPathInfos(int)	15	1	1	1	2	Superato
data::datamanager::DBHandler::readPathResult(int)	16	0	1	1	2	Superato
data::datamanager::DBHandler::readPathResults()	17	1	1	0	2	Superato
data::datamanager::DBHandler::readProof(int)	27	0	2	1	4	Superato
data::datamanager::DBHandler::readProofResults(int)	13	1	1	1	2	Superato
data::datamanager::DBHandler::readProximities(int)	12	1	1	1	2	Superato
data::datamanager::DBHandler::readProximity(int)	13	0	1	1	2	Superato
data::datamanager::DBHandler::readStep(int)	12	1	1	1	2	Superato
data::datamanager::DBHandler::readSteps(int)	14	1	1	1	2	Superato
data::datamanager::DBHandler::updateBuildings(Building[])	4	0	0	1	1	Superato
data::datamanager::DBHandler::updatePath(Path)	4	0	0	1	1	Superato
data::datamanager::DBHandler::updatePathInfo(int, PathInfo)	5	0	0	2	1	Superato
data::datamanager::DBHandler::updatePathResult(PathResult)	4	0	0	1	1	Superato
data::datamanager::DBHandler::updatePathResults(PathResult[])	4	0	0	1	1	Superato
data::datamanager::DBHandler::updateProofResult(int, ProofResult)	4	0	0	2	1	Superato
data::datamanager::DBHandler::updateStep(int, Step)	5	0	0	2	1	Superato
data::datamanager::DBHandler::writeBeacon(Beacon)	10	0	0	1	1	Superato
data::datamanager::DBHandler::writeBeacons(Beacon[])	5	1	1	1	2	Superato
data::datamanager::DBHandler::writeBuilding(int, Building)	23	0	0	2	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::datamanager::DBHandler::writeBuildings(Building[])	5	1	1	1	2	Superato
data::datamanager::DBHandler::writePath(Path)	11	0	0	1	1	Superato
data::datamanager::DBHandler::writePathInfo(int, PathInfo)	13	0	0	2	1	Superato
data::datamanager::DBHandler::writePathInfos(int, ArrayList <PathInfo>)	5	1	1	2	2	Superato
data::datamanager::DBHandler::writePathResult(PathResult)	11	0	0	1	1	Superato
data::datamanager::DBHandler::writePathResults(PathResult[])	5	1	1	1	2	Superato
data::datamanager::DBHandler::writePaths(Path[])	5	1	1	1	2	Superato
data::datamanager::DBHandler::writeProof(int, Proof)	11	0	0	2	1	Superato
data::datamanager::DBHandler::writeProofResult(int, ProofResult)	10	0	0	2	1	Superato
data::datamanager::DBHandler::writeProofResults(int, ArrayList <ProofResult>)	5	1	1	2	2	Superato
data::datamanager::DBHandler::writeProximities(int, ArrayList <Proximity>)	5	1	1	2	2	Superato
data::datamanager::DBHandler::writeProximity(int, int, Proximity)	10	0	0	3	1	Superato
data::datamanager::DBHandler::writeStep(int, int, Step)	14	0	0	3	1	Superato
data::datamanager::DBHandler::writeSteps(int, ArrayList <Step>)	5	1	1	2	2	Superato
data::datamanager::GetRankingDataRequest::getRemoteData(AbstractUrlRequestListener)	3	0	0	1	1	Superato
data::datamanager::GetRankingDataRequest::parseFromLocal()	1	0	0	0	1	Superato
data::datamanager::GetRankingDataRequest::parseFromUrlRequest(JSONObject)	14	1	2	1	3	Superato
data::datamanager::GetRankingDataRequest::updateLocalData(Score[])	1	0	0	1	1	Superato
data::datamanager::GetResultsDataRequest::getRemoteData(AbstractUrlRequestListener)	3	0	0	1	1	Superato
data::datamanager::GetResultsDataRequest::parseFromLocal()	3	0	0	0	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::datamanager::GetResultsDataRequest:: parseFromUrlRequest(JSONObject)	35	2	4	1	6	Superato
data::datamanager::GetResultsDataRequest:: updateLocalData(PathResult[])	3	0	0	1	1	Superato
data::datamanager::LoginManager:: change(String, String, String, AbstractDataManagerListener <Boolean>)	3	0	4	4	3	Superato
data::datamanager::LoginManager:: check(String, String, String, AbstractDataManagerListener <CheckResult[]>)	3	0	3	4	2	Superato
data::datamanager::LoginManager:: forgotPassword(String, AbstractDataManagerListener <Boolean>)	1	0	2	2	1	Superato
data::datamanager::LoginManager:: getLoggedUser()	3	0	0	0	1	Superato
data::datamanager::LoginManager:: getPro- fileData(AbstractDataManagerListener <Boolean>)	1	0	3	1	2	Superato
data::datamanager::LoginManager:: getToken()	6	0	1	0	2	Superato
data::datamanager::LoginManager:: isLoggedIn()	5	0	0	0	1	Superato
data::datamanager::LoginManager:: login(String, String, AbstractDataManagerListener <Boolean>)	3	0	3	3	3	Superato
data::datamanager::LoginManager:: logout(AbstractDataManagerListener <Boolean>)	8	0	2	1	1	Superato
data::datamanager::LoginManager:: registration(String, String, String, AbstractDataManagerListener <Boolean>)	3	0	4	4	5	Superato
data::datamanager::LoginManager:: sharedManager(Context)	5	0	0	1	2	Superato
data::datamanager::PathDataRequest:: ge- tRemoteData(AbstractUrlRequestListener)	3	0	0	1	1	Superato
data::datamanager::PathDataRequest:: parseFromLocal()	3	0	0	0	1	Superato
data::datamanager::PathDataRequest:: parseFromUrlRequest(JSONObject)	26	2	3	1	4	Superato
data::datamanager::PathDataRequest:: updateLocalData(Path)	3	0	0	1	1	Superato
data::datamanager::SaveDataManager:: execute()	0	0	0	0	0	Superato
data::datamanager::SaveDataManager:: returnError(ServerError)	3	0	0	1	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::datamanager::SaveDataManager::returnResponse()	3	0	0	0	1	Superato
data::datamanager::SaveResultDataRequest::execute()	3	0	2	0	1	Superato
data::LinearScoringAlgorithm::getScore(double, int, int)	6	0	0	3	1	Superato
data::MultipleChoiceTextQuiz::check(String)	3	0	0	1	1	Superato
data::MultipleChoiceTextQuiz::MultipleChoiceTextQuiz(String, String, String, String[])	5	0	0	4	1	Superato
data::Path:: equal(Beacon, RawBeacon)	3	0	0	2	3	Superato
data::Path:: searchProximity(RawBeacon)	9	2	2	1	4	Superato
data::PathProgress::addProofResult(ProofResult)	3	0	0	1	1	Superato
data::PathProgress:: getDuration()	4	0	0	0	1	Superato
data::PathProgress:: getEndTime()	3	0	0	0	1	Superato
data::PathProgress:: getPath()	3	0	0	0	1	Superato
data::PathProgress:: getProofResults()	3	0	0	0	1	Superato
data::PathProgress:: getStartTime()	3	0	0	0	1	Superato
data::PathProgress:: getTotalScore()	7	1	1	0	2	Superato
data::PathProgress:: setEndTime(GregorianCalendar)	3	0	0	1	1	Superato
data::ProofResult:: getDuration()	4	0	0	0	1	Superato
data::ScoringAlgorithmFactory::createScoringAlgorithm(JSONObject)	7	0	1	1	2	Superato
data::TestBuilder::createTest(JSONObject)	30	1	4	1	8	Superato
data::TestBuilder::getBiggerShapeTest(JSONObject)	13	1	2	1	3	Superato
data::TestBuilder::getMultipleChoiceTest(JSONObject)	25	2	3	1	6	Superato
data::TestBuilder::getTrueFalseTest(JSONObject)	18	1	3	1	4	Superato
data::TrueFalseTextQuiz:: check(boolean)	3	0	0	1	1	Superato
data::Utility::addNearestBuilding(Building, Building[], double, double)	15	1	2	4	3	Superato
data::Utility:: calculateTotalScore(List<ProofResult>)	8	1	1	1	2	Superato
data::Utility:: distanceToBuilding(Location, Building)	6	0	0	2	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
data::Utility:: insertBuilding(Building[], Building, int)	5	0	0	3	1	Superato
data::Utility:: stringToGregorianCalendar(String, String)	15	0	1	2	3	Superato
pathprogress::BeaconDiscoverDelegate:: didFoundBeacon(RawBeacon)	0	0	0	1	0	Superato
pathprogress::BeaconDiscoverDelegate:: didMoveFromBeacon(RawBeacon)	0	0	0	1	0	Superato
pathprogress::GPSListener:: onError(ServerError)	0	0	0	1	0	Superato
pathprogress::GPSListener:: onResponse(double, double)	0	0	0	2	0	Superato
pathprogress::GPSManager:: GPSManager(Context, GPSListener)	5	0	3	2	3	Superato
pathprogress:: PathProgressController:: addProofResult(ProofResult)	3	0	0	1	1	Superato
pathprogress:: PathProgressController:: didFoundBeacon(RawBeacon)	12	0	1	1	3	Superato
pathprogress:: PathProgressController:: didMoveFromBeacon(RawBeacon)	3	0	0	1	1	Superato
pathprogress:: PathProgressControllerDelegate:: didRangeProximity(Proximity)	0	0	0	1	0	Superato
pathprogress:: PathProgressControllerDelegate:: didReachStep(Step)	0	0	0	1	0	Superato
pathprogress::PathProgressMaker:: getCoordinates(Context, GPSListener)	3	0	0	2	1	Superato
urlrequest::AbstractUrlRequestListener:: onError(ServerError)	0	0	0	1	0	Superato
urlrequest::AbstractUrlRequestListener:: onResponse(JSONObject)	0	0	0	1	0	Superato
urlrequest::BuildingsRequest:: setBody(double, double, int, boolean)	14	0	1	4	3	Superato
urlrequest::ChangeProfileDataRequest:: setBody(String, String, String)	11	0	1	3	2	Superato
urlrequest::CheckRequest:: setBody(String, String, String)	11	0	1	3	2	Superato
urlrequest::ForgotPasswordRequest:: setBody(String)	9	0	1	1	2	Superato
urlrequest::GetRankingRequest:: setBody(int)	9	0	1	1	2	Superato
urlrequest::LoginRequest:: setBody(String, String)	10	0	1	2	2	Superato
urlrequest::RegistrationRequest:: setBody(String, String, String)	11	0	1	3	2	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
urlrequest::RequestMaker:: changeProfileData(Context, String, String, String, AbstractUrlRequestListener)	4	0	0	5	1	Superato
urlrequest::RequestMaker:: check(Context, String, String, String, AbstractUrlRequestListener)	4	0	0	5	1	Superato
urlrequest::RequestMaker:: forgotPassword(Context, String, AbstractUrlRequestListener)	4	0	0	3	1	Superato
urlrequest::RequestMaker:: getAppInfo(Context, AbstractUrlRequestListener)	4	0	0	2	1	Superato
urlrequest::RequestMaker:: getBuildings(Context, double, double, int, boolean, AbstractUrlRequestListener)	4	0	0	6	1	Superato
urlrequest::RequestMaker:: getPath(Context, int, AbstractUrlRequestListener)	4	0	0	3	1	Superato
urlrequest::RequestMaker:: getProfileData(Context, AbstractUrlRequestListener)	4	0	0	2	1	Superato
urlrequest::RequestMaker:: getRanking(Context, int, AbstractUrlRequestListener)	4	0	0	3	1	Superato
urlrequest::RequestMaker:: getResults(Context, AbstractUrlRequestListener)	4	0	0	2	1	Superato
urlrequest::RequestMaker:: login(Context, String, String, AbstractUrlRequestListener)	4	0	0	4	1	Superato
urlrequest::RequestMaker:: logout(Context, AbstractUrlRequestListener)	4	0	0	2	1	Superato
urlrequest::RequestMaker:: registration(Context, String, String, String, AbstractUrlRequestListener)	4	0	0	5	1	Superato
urlrequest::RequestMaker:: saveResult(Context, PathResult, AbstractUrlRequestListener)	4	0	0	3	1	Superato
urlrequest::SaveResultRequest:: setBody(PathResult)	27	1	2	1	4	Superato
urlrequest::URLRequest:: arrayRequest()	13	0	5	0	9	Superato
urlrequest::URLRequest:: execute(ResponseExpected)	13	0	1	1	3	Superato
urlrequest::URLRequest:: objectRequest()	16	0	5	0	8	Superato
urlrequest::URLRequest:: signalError()	1	0	0	0	1	Superato
viewController::authentication:: AccountActivity::onCreate(Bundle)	13	0	1	1	2	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
viewController::authentication::AccountActivity::setButtonInfoProfilo()	5	0	3	0	2	Superato
viewController::authentication::ConfirmRegistration::onBackPressed()	4	0	0	0	1	Superato
viewController::authentication::ConfirmRegistration::onCreate(Bundle)	13	0	0	1	1	Superato
viewController::authentication::ConfirmRegistration::setButton()	5	0	3	0	2	Superato
viewController::authentication::ForgotPassword::onCreate(Bundle)	6	0	0	1	1	Superato
viewController::authentication::ForgotPassword::setButton()	4	0	3	0	2	Superato
viewController::authentication::LoginActivity::checkFields(String, String)	17	0	1	2	5	Superato
viewController::authentication::LoginActivity::onCreate(Bundle)	11	0	0	1	1	Superato
viewController::authentication::LoginActivity::setButton()	5	0	3	0	2	Superato
viewController::authentication::LoginActivity::setLoginButton()	4	0	7	0	4	Superato
viewController::authentication::RecoverPassword::onCreate(Bundle)	6	0	0	1	1	Superato
viewController::authentication::RegistrationActivity::checkFields()	30	0	3	0	9	Superato
viewController::authentication::RegistrationActivity::clearFields()	11	0	0	0	1	Superato
viewController::authentication::RegistrationActivity::login(String, String, String)	4	0	3	3	2	Superato
viewController::authentication::RegistrationActivity::onCreate(Bundle)	14	0	0	1	1	Superato
viewController::authentication::RegistrationActivity::setButton()	4	0	6	0	4	Superato
viewController::building::BuildingActivity::onCreate(Bundle)	4	1	5	1	5	Superato
viewController::building::BuildingActivity::setButtons()	5	0	3	0	2	Superato
viewController::building::BuildingActivity::setItems()	13	0	2	0	1	Superato
viewController::building::BuildingSearchActivity::onCreate(Bundle)	20	0	0	1	1	Superato
viewController::building::BuildingSearchActivity::setButton()	2	1	9	0	7	Fallito
viewController::building::BuildingSearchActivity::setCheckBoxSignal()	10	0	3	0	2	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
viewController::building:: BuildingSearchActivity::setItems()	7	0	2	0	1	Superato
viewController::building:: BuildingSearchActivity::setSeekBarSignal()	4	0	2	0	1	Superato
viewController::building:: PathActivity::onCreate(Bundle)	17	0	0	1	1	Superato
viewController::building:: PathActivity::setButton()	5	0	3	0	2	Superato
viewController::games:: MultipleChoiceQuizActivity::onCreate(Bundle)	5	0	3	1	2	Superato
viewController::games:: ProofActivity::onCreate(Bundle)	5	0	0	1	1	Superato
viewController::games:: ProofResultActivity::onCreate(Bundle)	5	0	0	1	1	Superato
viewController::games:: SearchNewStepActivity::configureFilters()	3	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::configureListeners()	4	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::configureProximityManager()	7	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::configureSpaces()	7	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::createBeaconListener()	9	0	3	0	2	Superato
viewController::games:: SearchNewStepActivity::createScanStatusListener()	4	0	2	0	1	Superato
viewController::games:: SearchNewStepActivity::onCreate(Bundle)	22	0	0	1	1	Superato
viewController::games:: SearchNewStepActivity::onDestroy()	6	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::onStart()	5	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::onStop()	5	0	0	0	1	Superato
viewController::games:: SearchNewStepActivity::setButtons()	5	0	3	0	2	Superato
viewController::games:: SearchNewStepActivity::showToast(String)	4	0	2	1	1	Superato
viewController::games:: SearchNewStepActivity::startScanning()	4	0	2	0	1	Superato
viewController::games:: TrueFalseQuiz::onCreate(Bundle)	5	0	0	1	1	Superato
viewController::savedresults:: RankingActivity::onCreate(Bundle)	9	0	0	1	1	Superato
viewController::savedresults:: ResultActivity::onCreate(Bundle)	6	0	0	1	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
viewController::savedresults::SavedResultsActivity::getListData()	12	0	0	0	1	Superato
viewController::savedresults::SavedResultsActivity::onCreate(Bundle)	19	0	1	1	2	Superato
viewController::utility::AppInfoActivity::onCreate(Bundle)	3	0	2	1	1	Superato
viewController::utility::BuildingAdapter::getCount()	3	0	0	0	1	Superato
viewController::utility::BuildingAdapter::getItem(int)	3	0	0	1	1	Superato
viewController::utility::BuildingAdapter::getItemId(int)	3	0	0	1	1	Superato
viewController::utility::BuildingAdapter::getView(int, View, ViewGroup)	18	0	1	3	2	Superato
viewController::utility::MenuActivity::getDrawerToggleDelegate()	5	0	0	0	1	Superato
viewController::utility::MenuActivity::logout()	4	0	3	0	2	Superato
viewController::utility::MenuActivity::selectDrawerItem(MenuItem)	31	0	1	1	8	Superato
viewController::utility::MenuActivity::setContentView(int)	24	0	1	1	3	Superato
viewController::utility::MenuActivity::setUpContent(NavigationView)	5	0	2	1	2	Superato
viewController::utility::MenuActivity::updateView()	17	0	1	0	2	Superato
viewController::utility::MenuActivity::useDrawerToggle()	1	0	0	0	1	Superato
viewController::utility::MenuActivity::useToolbar()	4	0	0	0	1	Superato
viewController::utility::ResultsAdapter::getCount()	4	0	0	0	1	Superato
viewController::utility::ResultsAdapter::getItem(int)	4	0	0	1	1	Superato
viewController::utility::ResultsAdapter::getItemId(int)	4	0	0	1	1	Superato
viewController::utility::ResultsAdapter::getView(int, View, ViewGroup)	21	0	1	3	2	Superato
viewController::utility::risultatoProva::getData()	3	0	0	0	1	Superato
viewController::utility::risultatoProva::getDurata()	3	0	0	0	1	Superato
viewController::utility::risultatoProva::getEdificio()	3	0	0	0	1	Superato
viewController::utility::risultatoProva::getPunteggio()	3	0	0	0	1	Superato

Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
viewController::utility::risultatoProva::setData(String)	3	0	0	1	1	Superato
viewController::utility::risultatoProva::setDurata(String)	3	0	0	1	1	Superato
viewController::utility::risultatoProva::setEdificio(String)	3	0	0	1	1	Superato
viewController::utility::risultatoProva::setPunteggio(String)	3	0	0	1	1	Superato
viewController::utility::SplashActivity::onCreate(Bundle)	6	0	2	1	1	Superato
clips::MainActivity:: onCreate(Bundle)	8	0	0	1	1	Superato

A.7.0.3.4 Server

::Package::Classe::Metodo	LC	LA	CI	NP	CC	Esito
urlrequesthandler::EmailChecker::checkEmail	19	0	0	1	3	Superato
urlrequesthandler::PasswordChecker::isValid	10	1	0	1	7	Superato
urlrequesthandler::DBHandler::getDB	2	0	0	0	3	Superato
urlrequesthandler::BuildingsProvider::addPAT	18	1	1	1	2	Superato
urlrequesthandler::LoginHandler::userID	18	0	0	2	2	Superato
urlrequesthandler::PathProvider::fulfilledProo	15	0	0	1	2	Superato
urlrequesthandler::PathProvider::fulfilledProx	18	0	0	1	2	Superato
urlrequesthandler::PathProvider::fulfilledStep	17	0	0	1	2	Superato
urlrequesthandler::PathProvider::getAlgorith	6	0	0	1	2	Superato
urlrequesthandler::PathProvider::getBeacon	13	0	0	1	2	Superato
urlrequesthandler::PathProvider::getPath	19	0	0	1	2	Superato
urlrequesthandler::PathProvider::getProof	16	0	0	1	2	Superato
urlrequesthandler::PathProvider::getProximit	18	0	0	1	2	Superato
urlrequesthandler::PathProvider::getSteps	15	0	0	1	2	Superato
urlrequesthandler::PathProvider::getTest	6	0	0	1	2	Superato
urlrequesthandler::BuildingsProvider::addDist	2	0	0	1	1	Superato
urlrequesthandler::URLRequestHandler::exec	12	0	0	0	1	Superato

B Specifica dei test

B.1 Descrizione dei test

Vengono ora indicati i test di validazione, di sistema e di integrazione previsti e infine i test di unità.

Per ogni test ci sarà la dicitura N.I. qualora non fossero ancora stati implementati altrimenti IMPL. Per la sintassi con cui verranno esposti si rimanda al documento “*Norme di Progetto v4.0.0*”.

B.2 Modello a V

Per la specificazione dei test si utilizzerà il Modello a V. Secondo questo modello il testing del software viene suddiviso in livelli differenti, i quali si concretizzano in un'esecuzione bottom-up che avanza sequenzialmente alle attività di codifica e di validazione. Ad ogni livello corrisponde un ciclo di uno specifico tipo di test, ed ogni test viene creato in base al relativo livello di progettazione.

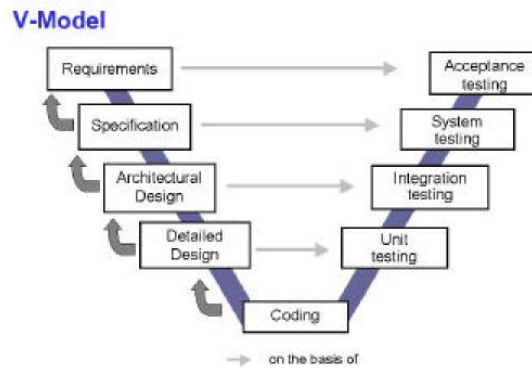


Figura 1: Modello a V

B.3 Test di validazione

I test di validazione servono per accertarsi che il prodotto realizzato sia conforme alle attese di Miriade SpA.

Per ognuno vengono indicati i passi necessari all'utente per testare i requisiti associati. Il tracciamento tra i test di validazione e i requisiti correlati viene riportato nel documento “*Analisi dei Requisiti v4.0.0*”.

Test	Descrizione	Operazioni
TV0F1	L'utente vuole verificare di poter possedere un account.	Viene richiesto di: 1. registrare un account; 2. autenticarsi con i dati del proprio account.

Test	Descrizione	Operazioni
TV0F1.1	L'utente vuole verificare che si possa autenticare.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire un indirizzo email nel campo apposito; 2. inserire una password nel campo apposito; 3. confermare i dati inseriti; 4. verificare di essere autenticato se i dati inseriti sono corretti; 5. verificare se viene segnalato ogni eventuale errore durante la procedura di autenticazione.
TV0F1.1.1	L'utente vuole verificare che l'autenticazione richieda l'inserimento di una mail.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire alcuna mail; 2. verificare che il sistema faccia visualizzare l'errore relativo.
TV0F1.1.2	L'utente vuole verificare che l'autenticazione richieda una password.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire alcuna password; 2. verificare che il sistema faccia visualizzare l'errore relativo.
TV0F1.1.3	L'utente vuole verificare che si possa confermare l'autenticazione.	Viene richiesto di: <ol style="list-style-type: none"> 1. cliccare sul pulsante di conferma; 2. verificare che l'autenticazione avvenga correttamente se i dati sono corretti, altrimenti vengano mostrati gli errori.
TV0F1.1.4	L'utente vuole verificare che se i dati sono corretti egli viene autenticato.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire i dati corretti richiesti dall'autenticazione; 2. confermare i dati cliccando l'apposito pulsante; 3. verificare di essere autenticato.
TV0F1.1.5	L'utente vuole verificare che, ogni eventuale errore riguardante la procedura di autenticazione, interrompa la procedura stessa.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire i dati non corretti richiesti dall'autenticazione; 2. confermare l'autenticazione cliccando l'apposito bottone; 3. verificare che venga visualizzato il messaggio di errore di autenticazione.

Test	Descrizione	Operazioni
TV0F1.1.5.1	L'utente vuole verificare che il sistema interrompa l'autenticazione se la mail inserita non è presente nel sistema.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire una mail non presente nel database dell'applicazione; 2. confermare l'autenticazione cliccando l'apposito bottone; 3. verificare che venga mostrato l'errore di autenticazione.
TV0F1.1.5.2	L'utente vuole verificare che il sistema interrompa l'autenticazione se la password inserita non coincide con quella associata alla mail.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire una mail valida; 2. inserire una password errata; 3. confermare l'autenticazione cliccando l'apposito tasto; 4. verificare che venga mostrato l'errore di autenticazione.
TV0F1.1.5.3	L'utente vuole verificare che il sistema mostri l'errore di autenticazione.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire i dati errati richiesti dall'autenticazione; 2. confermare cliccando l'apposito bottone; 3. verificare che venga visualizzato l'errore di autenticazione.
TV0F1.1.5.4	L'utente vuole verificare di poter cambiare la password nel caso non se la ricordasse.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire l'indirizzo email con cui si è registrato; 2. verificare di aver ricevuto un'email all'indirizzo inserito con una nuova password casuale; 3. verificare che la vecchia password del proprio account venga sostituita con quella nuova inviata per email.
TV0F1.1.5.4.1	L'utente vuole verificare che la richiesta di una nuova password richieda una mail.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire una mail; 2. confermare la richiesta di una nuova password cliccando l'apposito bottone; 3. verificare che il sistema mostri l'errore di inserimento del dato.
TV0F1.1.5.4.2	L'utente vuole verificare che la richiesta di una nuova password invii la nuova password al suo contatto mail.	Viene richiesto di: <ol style="list-style-type: none"> 1. completare la richiesta di una nuova password; 2. verificare sulla propria casella di posta che sia stata inviata la nuova password.

Test	Descrizione	Operazioni
TV0F1.1.5.4.3	L'utente vuole verificare che il sistema abbia cambiato la password con la nuova password inviata.	Viene richiesto di: <ol style="list-style-type: none"> 1. effettuare l'autenticazione utilizzando la nuova password; 2. verificare che l'autenticazione vada a buon fine.
TV0F1.1.5.4.4	L'utente vuole verificare che il sistema interrompa la richiesta di nuova password se la mail inserita non è presente nel database.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire la mail corretta; 2. confermare la richiesta di nuova password cliccando l'apposito bottone; 3. verificare che il sistema interrompa la procedura mostrando l'errore apposito.
TV0F1.1.5.4.5	L'utente vuole verificare che il sistema segnali l'errore di mail non presente nel database.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire una mail non presente nel database; 2. confermare la richiesta di nuova password cliccando l'apposito bottone; 3. verificare che venga visualizzato l'errore di mail non presente.
TV0F1.2	L'utente vuole verificare di poter registrare un nuovo account.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire il proprio indirizzo email nell'apposito campo; 2. inserire uno username nell'apposito campo; 3. inserire una password nell'apposito campo; 4. reinserire la password nell'apposito campo; 5. confermare i dati inseriti cliccando l'apposito bottone; 6. verificare che l'account venga registrato con i dati inseriti, se non ci sono stati errori, e che venga autenticato automaticamente.
TV0F1.2.1	L'utente vuole verificare che la registrazione richieda la mail.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire alcuna mail; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione mostri l'errore di mail mancante.

Test	Descrizione	Operazioni
TV0F1.2.2	L'utente vuole verificare che la registrazione richieda un username.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire alcun username; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione mostri l'errore di username mancante.
TV0F1.2.3	L'utente vuole verificare che la registrazione richieda la password.	Viene richiesto di: <ol style="list-style-type: none"> 1. non inserire alcuna password; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione mostri l'errore di password mancante.
TV0F1.2.4	L'utente vuole verificare che la registrazione richieda di reinserire la password.	Viene richiesto di: <ol style="list-style-type: none"> 1. non reinserire la password; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione mostri l'errore di reinserimento della password errato.
TV0F1.2.5	L'utente vuole verificare che la registrazione debba essere confermata tramite l'apposito bottone.	Viene richiesto di: <ol style="list-style-type: none"> 1. completare i campi richiesti dalla registrazione; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che, se i dati sono corretti, si venga registrati, altrimenti vengano mostrati gli errori relativi ai dati.
TV0F1.2.6	L'utente vuole verificare che se i dati inseriti sono corretti e si conferma la registrazione, egli viene registrato.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire tutti i dati richiesti dalla registrazione in modo corretto; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione vada a buon fine.
TV0F1.2.6.1	L'utente vuole verificare che se la registrazione va a buon fine, egli viene automaticamente autenticato.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire i dati richiesti in modo corretto; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare di essere autenticati non appena la registrazione conclude con esito positivo.

Test	Descrizione	Operazioni
TV0F1.2.7	L'utente vuole verificare che il sistema avvisi l'utente che si sta registrando, che durante la procedura di registrazione è avvenuto un errore.	Viene richiesto di: <ol style="list-style-type: none"> 1. completare i campi richiesti dalla registrazione; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che sia mostrato un avviso qualora ci siano stati errori.
TV0F1.2.7.1.1	L'utente vuole verificare che una mail valida sia formata dal simbolo '@' preceduto da caratteri e seguita da un dominio valido.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire una mail che non rispetti i criteri per essere valida; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e venga mostrato l'errore di mail non valida.
TV0F1.2.7.1.2	L'utente vuole verificare che la mail inserita non debba essere già in uso.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire una mail già in uso; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e venga mostrato l'errore di mail già in uso.
TV0F1.2.7.2	L'utente vuole verificare che l'username richiesto debba essere valido.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire un username non valido; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e sia mostrato l'avviso di username non valido.
TV0F1.2.7.2.1	L'utente vuole verificare che un username valido sia formato da solo caratteri alfanumerici.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire un username che non rispetti i criteri per essere valido; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e venga mostrato l'errore di username non valido.

Test	Descrizione	Operazioni
TV0F1.2.7.2.2	L'utente vuole verificare che l'username inserito non debba essere già in uso.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire un'username già in uso; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e venga mostrato l'errore di username già in uso.
TV0F1.2.7.3	L'utente vuole verificare che la password deve avere un minimo di 6 caratteri e un massimo di 16.	Viene richiesto di: <ol style="list-style-type: none"> 1. inserire una password più corta o più lunga di quello stabilito; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e venga mostrato l'errore di password non valida.
TV0F1.2.7.4	L'utente vuole verificare che la password reinserita debba essere uguale a quella precedentemente inserita.	Viene richiesto di: <ol style="list-style-type: none"> 1. reinserire una password diversa da quella precedentemente inserita; 2. confermare la registrazione cliccando l'apposito bottone; 3. verificare che la registrazione venga interrotta e venga mostrato l'errore di password reinserita non coincidente.
TV0F2	L'utente vuole verificare di poter fare il logout.	Viene richiesto di: <ol style="list-style-type: none"> 1. cliccare il bottone per effettuare il logout; 2. verificare che sia stato effettuato il logout.
TV0F3	L'utente vuole verificare di poter effettuare un percorso se si trova in un luogo con dei percorsi disponibili.	Viene richiesto di: <ol style="list-style-type: none"> 1. trovarsi in un luogo con dei percorsi disponibili; 2. cliccare sul bottone per iniziare il percorso; 3. verificare di poter svolgere il percorso seguendo le indicazioni.

Test	Descrizione	Operazioni
TV0F3.1	L'utente vuole verificare di poter giocare il percorso fino alla sua conclusione.	Viene richiesto di: <ol style="list-style-type: none"> 1. iniziare il percorso cliccando sul bottone di inizio percorso; 2. svolgere il percorso seguendo le indicazioni; 3. verificare di poter svolgere il percorso fino alla sua conclusione.
TV0F3.1.1	L'utente vuole verificare di poter cercare e trovare il beacon della prima stazione.	Viene richiesto di: <ol style="list-style-type: none"> 1. muoversi all'interno dell'edificio per cercare il beacon della prima stazione; 2. verificare che sia possibile cercare il beacon della prima stazione; 3. verificare che, se si è nei paraggi del beacon della prima stazione, si possa trovarlo.
TV0F3.1.1.1	L'utente vuole verificare che se il dispositivo trova il beacon della stazione ricercata egli può iniziare giocare la prova.	Viene richiesto di: <ol style="list-style-type: none"> 1. recarsi al beacon della stazione che sta cercando; 2. verificare di poter iniziare a giocare la prova della stazione cliccando l'apposito bottone di inizio prova.
TV0F3.1.1.2	L'utente vuole verificare che il dispositivo ignori i beacon che non sono quello che si sta cercando.	Viene richiesto di: <ol style="list-style-type: none"> 1. recarsi presso un beacon che non sia quello che si sta cercando; 2. verificare che il dispositivo ignori il beacon.
TV0F3.1.2	L'utente vuole verificare di poter svolgere la prova della stazione in cui si trova.	Viene richiesto di: <ol style="list-style-type: none"> 1. trovarsi nella stazione giusta; 2. verificare di poter iniziare e concludere di giocare la prova della stazione in cui si trova.
TV0F3.1.2.1	L'utente vuole verificare che la prova che viene proposta sia una di quelle previste.	Viene richiesto di: <ol style="list-style-type: none"> 1. trovarsi nella stazione esatta; 2. iniziare la prova della stazione cliccando il bottone di inizio prova; 3. verificare che il sistema somministri una prova corretta.

Test	Descrizione	Operazioni
TV0F3.1.2.2	L'utente vuole verificare che gli vengano mostrate le istruzioni per lo svolgimento della prova.	Viene richiesto di: <ol style="list-style-type: none"> 1. trovarsi nella stazione corretta; 2. verificare che vengano mostrate le istruzioni per lo svolgimento della prova.
TV0F3.1.2.3	L'utente vuole verificare di poter giocare la prova.	Viene richiesto di: <ol style="list-style-type: none"> 1. iniziare la prova cliccando sul bottone di inizio prova; 2. verificare di poter giocare la prova fino alla sua conclusione.
TV0F3.1.2.4	L'utente vuole verificare che al termine della prova venga mostrato il risultato ottenuto.	Viene richiesto di: <ol style="list-style-type: none"> 1. giocare la prova fino alla sua conclusione; 2. verificare che una volta conclusa la prova venga mostrato il risultato ottenuto.
TV0F3.1.2.5	L'utente vuole verificare che una volta conclusa la prova sia possibile continuare con lo svolgimento del percorso.	Viene richiesto di: <ol style="list-style-type: none"> 1. giocare la prova fino alla sua conclusione; 2. verificare di poter continuare lo svolgimento del percorso una volta conclusa la prova.
TV0F3.1.2.5.1	L'utente vuole verificare che, se la prova non è l'ultima del percorso, vengano mostrate le istruzioni per trovare la prossima stazione.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere una prova che non sia l'ultima del percorso; 2. verificare che vengano mostrate le istruzioni per trovare la prossima stazione.
TV0F3.1.2.5.1.1	L'utente vuole verificare che una volta conclusa la prova della stazione in cui si trova, vengano mostrate le istruzioni per trovare la stazione seguente.	Viene richiesto di: <ol style="list-style-type: none"> 1. giocare la prova fino alla sua conclusione; 2. verificare che vengano mostrate le istruzioni per la ricerca della nuova stazione.
TV0F3.1.2.5.2	L'utente vuole verificare che se la prova è l'ultima del percorso egli ha concluso il percorso.	Viene richiesto di: <ol style="list-style-type: none"> 1. completare tutte le prove; 2. verificare che una volta finita l'ultima prova sia abbia concluso anche il percorso.
TV0F3.2	L'utente vuole verificare di poter visualizzare correttamente i risultati del percorso ottenuti una volta che ha concluso tutte le prove.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove; 2. verificare che vengano mostrati i risultati ottenuti nel percorso.

Test	Descrizione	Operazioni
TV0F3.2.1	L'utente vuole verificare che, una volta concluso il percorso, venga mostrato il tempo totale che egli ci ha messo a completarlo.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove; 2. verificare che venga mostrato il tempo totale per completare il percorso.
TV0F3.2.1.1	L'utente vuole verificare che il tempo totale venga calcolato dal momento in cui si accetta di giocare il percorso a quando si conclude l'ultima prova.	Viene richiesto di: <ol style="list-style-type: none"> 1. annotarsi l'ora di inizio dello svolgimento del percorso; 2. concludere tutte le prove del percorso; 3. annotarsi l'ora di conclusione del percorso; 4. verificare che il tempo mostrato dall'applicazione sia uguale al tempo calcolato con le rilevazioni fatte.
TV0F3.2.2	L'utente vuole verificare che, una volta concluso il percorso, venga mostrato il tempo totale che egli ha impiegato per completare tutte le prove.	Viene richiesto di: <ol style="list-style-type: none"> 1. completare tutte le prove; 2. verificare che venga mostrato il tempo totale per completare tutte le prove.
TV0F3.2.2.1	L'utente vuole verificare che il tempo totale per svolgere tutte le prove sia calcolato sommando i tempi per concludere le varie prove (senza quindi contare il tempo per gli spostamenti da stazione a stazione).	Viene richiesto di: <ol style="list-style-type: none"> 1. completare le varie prove annotandosi i tempi per svolgerle; 2. verificare che il tempo mostrato sia la somma dei tempi per svolgere le varie prove.
TV0F3.2.2.1.1	L'utente vuole verificare che il tempo per svolgere una prova sia calcolato dal momento in cui si accetta di affrontarla a quando la si conclude. All'utente viene richiesto di:	Viene richiesto di: <ol style="list-style-type: none"> 1. annotare il tempo di inizio della prova; 2. annotare il tempo di conclusione della prova; 3. verificare che il tempo venga calcolato dal tempo trascorso dall'inizio alla fine della prova.
TV0F3.2.3	L'utente vuole verificare che venga mostrato il punteggio totale una volta concluso il percorso.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove; 2. verificare che venga mostrato il punteggio totale del percorso.

Test	Descrizione	Operazioni
TV0F3.2.3.1	L'utente vuole verificare che il punteggio totale venga calcolato sommando i punteggi delle prove.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove e annotare i punteggi delle prove; 2. verificare che il punteggio totale sia dato dalla somma dei punteggi di tutte le prove.
TV0F3.2.4	L'utente vuole verificare che venga mostrata la posizione in classifica con il relativo punteggio.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove del percorso; 2. verificare che venga mostrata la posizione in classifica con il relativo punteggio.
TV0F3.2.4.1	L'utente vuole verificare di poter visualizzare la classifica generale.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove del percorso; 2. verificare di poter visualizzare la classifica generale.
TV0F3.2.5	L'utente vuole verificare che venga mostrata la prova in cui ha ottenuto il punteggio più alto.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove del percorso; 2. verificare che venga mostrata la prova in cui ha ottenuto il miglior punteggio.
TV0F3.2.6	L'utente vuole verificare che venga mostrata la prova in cui ha ottenuto il punteggio più basso.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove del percorso; 2. verificare che venga mostrata la prova in cui ha ottenuto il peggior punteggio.
TV0F3.3	L'utente vuole verificare che, se è autenticato, può salvare i risultati del percorso.	Viene richiesto di: <ol style="list-style-type: none"> 1. concludere tutte le prove del percorso; 2. verificare di poter salvare i risultati del percorso, cliccando sull'apposito bottone, se già autenticato.
TV0F4	L'utente vuole verificare di poter visualizzare le informazioni relative all'app.	Viene richiesto di: <ol style="list-style-type: none"> 1. cliccare sulla sezione riguardante le informazioni relative all'app tramite l'apposito bottone; 2. verificare di poter visualizzare le informazioni.

Test	Descrizione	Operazioni
TV0F4.1	L'utente vuole verificare che venga mostrata una descrizione generale dell'app.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante le informazioni relative all'app tramite l'apposito bottone; 2. verificare che venga visualizzata una descrizione generale dell'app.
TV0F4.2	L'utente vuole verificare di poter accedere alla pagina web relativa all'app tramite un link.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante le informazioni relative all'app tramite l'apposito bottone; 2. verificare che sia presente un link alla pagina web relativa all'app e questa si apra tramite il browser predefinito del dispositivo.
TV0F4.3	L'utente vuole verificare di poter inviare una segnalazione di errore tramite mail.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante le informazioni relative all'app tramite l'apposito bottone; 2. verificare che sia presente l'indirizzo mail per la segnalazione di errori.
TV0F4.3.1	L'utente vuole verificare che, cliccando sull'indirizzo mail per la segnalazione di errori, venga aperto il gestore delle mail predefinito del dispositivo e che il destinatario sia quello su cui a cliccato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante le informazioni relative all'app tramite l'apposito bottone; 2. cliccare sull'indirizzo mail per la segnalazione degli errori; 3. verificare che si apra il gestore delle mail predefinito nel dispositivo e che il destinatario sia effettivamente quello per le segnalazioni d'errore.
TV0F5	L'utente vuole verificare di poter visualizzare i risultati dei percorsi effettuati precedentemente.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. verificare di poter visualizzare i risultati salvati qualora sia autenticato.

Test	Descrizione	Operazioni
TV0F5.1	L'utente vuole verificare di poter visualizzare la lista dei percorsi salvati qualora sia autenticato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. verificare che venga mostrata la lista dei suoi risultati salvati.
TV0F5.1.1	L'utente vuole verificare di poter visualizzare tutte le informazioni relative al percorso salvato scelto.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che vengano mostrate tutte le informazioni relative al percorso selezionato.
TV0F5.1.1.1	L'utente vuole verificare che venga visualizzato il nome del percorso selezionato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che venga visualizzato il nome del percorso.
TV0F5.1.1.2	L'utente vuole verificare che venga visualizzato il nome del luogo dove è stato svolto il percorso selezionato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che venga visualizzato il nome del luogo in cui è stato svolto il percorso.

Test	Descrizione	Operazioni
TV0F5.1.1.3	L'utente vuole verificare che venga visualizzata la data in cui si è svolto il percorso selezionato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che venga visualizzata la data in cui è stato svolto il percorso.
TV0F5.1.1.4	L'utente vuole verificare che venga visualizzato il punteggio totale ottenuto nel percorso selezionato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che venga visualizzato il punteggio totale ottenuto nel percorso.
TV0F5.1.1.5	L'utente vuole verificare che venga visualizzato il tempo totale impiegato per svolgere il percorso selezionato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che venga visualizzato il tempo totale impiegato per svolgere il percorso.
TV0F5.1.1.6	L'utente vuole verificare che venga visualizzato la posizione in classifica del percorso selezionato.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. selezionare un risultato tra quelli salvati; 3. verificare che venga visualizzato la posizione in classifica del percorso.

Test	Descrizione	Operazioni
TV0F5.2	L'utente vuole verificare che se è autenticato ma non ha ancora salvato percorsi venga visualizzato un avviso che inviti a svolgerne almeno uno.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 2. verificare che venga visualizzato un avviso che solleciti a svolgere e salvare almeno un percorso.
TV0F5.3	L'utente vuole verificare che se non è autenticato viene visualizzato un messaggio che invita ad autenticarsi.	Viene richiesto di: <ol style="list-style-type: none"> 1. non effettuare il login; 2. accedere alla sezione riguardante i risultati dei percorsi tramite l'apposito bottone; 3. verificare che venga visualizzato un avviso che inviti ad autenticarsi.
TV0F6	L'utente vuole verificare di poter visualizzare gli edifici con dei percorsi disponibili più vicini alla sua posizione.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. verificare di poter visualizzare gli edifici con percorsi più vicini.
TV0F6.1.1	L'utente vuole verificare di poter inserire un raggio in chilometri per la ricerca degli edifici.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. verificare di poter inserire un raggio per la ricerca in chilometri.
TV0F6.1.1.1	L'utente vuole verificare che se la ricerca tramite raggio va a buon fine vengano visualizzati gli edifici entro quel raggio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. inserire un raggio nel quale sia presente un edificio con percorsi; 3. verificare che venga visualizzato l'edificio.

Test	Descrizione	Operazioni
TV0F6.2	L'utente vuole verificare di poter visualizzare tutte le informazioni relative all'edificio scelto tra la lista di ricerca degli edifici.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli nella lista; 3. verificare che vengano visualizzate le informazioni relative a quell'edificio.
TV0F6.2.1	L'utente vuole verificare di poter visualizzare tutte le informazioni specifiche dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli nella lista; 3. verificare di poter visualizzare delle informazioni specifiche dell'edificio.
TV0F6.2.1.1	L'utente vuole verificare che venga visualizzato il nome dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare che venga visualizzato il nome dell'edificio.
TV0F6.2.1.2	L'utente vuole verificare che venga visualizzata la destinazione d'uso dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare che venga visualizzata la destinazione d'uso dell'edificio.

Test	Descrizione	Operazioni
TV0F6.2.1.3	L'utente vuole verificare che venga visualizzato l'indirizzo dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare che venga visualizzato l'indirizzo dell'edificio.
TV0F6.2.2	L'utente vuole verificare che venga visualizzato il link al sito web dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare che venga visualizzato il link al sito web dell'edificio.
TV0F6.2.3	L'utente vuole verificare di poter visualizzare tutti i contatti disponibili dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare di poter visualizzare i contatti disponibili dell'edificio.
TV0F6.2.3.1	L'utente vuole verificare di poter visualizzare il contatto telefonico dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare di poter visualizzare il contatto telefonico dell'edificio.

Test	Descrizione	Operazioni
TV0F6.2.3.2	L'utente vuole verificare di poter visualizzare il contatto mail dell'edificio.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare di poter visualizzare il contatto mail dell'edificio.
TV0F6.2.3.3	L'utente vuole verificare di poter contattare contattare, tramite la pagina Facebook, l'edificio, qualora sia disponibile.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare, qualora sia disponibile, di poter contattare l'edificio tramite la pagina Facebook.
TV0F6.2.3.4	L'utente vuole verificare di poter contattare contattare, tramite Twitter, l'edificio, qualora sia disponibile.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare, qualora sia disponibile, di poter contattare l'edificio tramite Twitter.
TV0F6.2.3.5	L'utente vuole verificare di poter contattare contattare, tramite contatto di WhatsApp, l'edificio, qualora sia disponibile.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare, qualora sia disponibile, di poter contattare l'edificio tramite il contatto di WhatsApp.

Test	Descrizione	Operazioni
TV0F6.2.3.6	L'utente vuole verificare di poter contattare contattare, tramite contatto Telegram, l'edificio, qualora sia disponibile.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere alla sezione riguardante la ricerca degli edifici tramite l'apposito bottone; 2. selezionare un edificio tra quelli della lista; 3. verificare, qualora sia disponibile, di poter contattare l'edificio tramite il contatto di Telegram.
TV0F7	L'utente verifica di poter cambiare le proprie credenziali d'accesso.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere al proprio profilo tramite l'apposito bottone; 2. verificare di poter cambiare le proprie credenziali.
TV0F7.1	L'utente vuole verificare di poter modificare il proprio username.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere al proprio profilo tramite l'apposito bottone; 2. modificare il proprio username rispettando i requisiti R0F1.2.7.2.1 e R0F1.2.7.2.2; 3. confermare il cambio delle credenziali cliccando l'apposito bottone; 4. verificare di aver modificato il proprio username.
TV0F7.2	L'utente vuole verificare di poter modificare la propria password.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere al proprio profilo tramite l'apposito bottone; 2. modificare la propria password rispettando il requisito R0F1.2.7.3; 3. confermare il cambio delle credenziali cliccando l'apposito bottone; 4. verificare di aver modificato la propria password.

Test	Descrizione	Operazioni
TV0F7.3	L'utente vuole verificare che se i dati inseriti non sono corretti vengono visualizzati gli errori relativi.	Viene richiesto di: <ol style="list-style-type: none"> 1. accedere al proprio profilo tramite l'apposito bottone; 2. modificare i propri dati con dati che non rispettano i requisiti R0F1.2.7.3, R0F1.2.7.2.1 e R0F1.2.7.2.2; 3. confermare il cambio delle credenziali cliccando l'apposito bottone; 4. verificare che venga interrotta la procedura di cambio credenziali e che vengano visualizzati gli errori relativi ai dati.
TV0Q1	L'utente verifica se viene fornito il manuale per l'utente dell'applicazione.	Viene richiesto di: <ol style="list-style-type: none"> 1. verificare se viene fornito il manuale per l'utente dell'applicazione.

Tabella 20: Riepilogo test di validazione

B.4 Test di sistema

I test di sistema servono per accertarsi che il comportamento dinamico del sistema rispetti i requisiti software individuati e descritti nel documento “Analisi dei Requisiti v4.0.0”.

B.4.1 Descrizione dei test di sistema

Test	Descrizione	Stato	Requisito
TS0F1	Viene verificato che il sistema permetta la creazione e la gestione di un account	N.I	R0F1
TS0F2	Viene verificato che il sistema permetta la deautenticazione dell'utente	N.I	R0F2
TS0F3	Viene verificato che il sistema permetta all'utente di giocare un percorso tra quelli disponibili nel luogo in cui si trova	N.I	R0F3
TS0F4	Viene verificato che il sistema permetta all'utente di visualizzare le informazioni dell'app	N.I	R0F4
TS0F5	Viene verificato che il sistema permetta all'utente di visualizzare i risultati dei percorsi effettuati precedentemente	N.I	R0F5
TS0F6	Viene verificato che il sistema permetta all'utente di cercare quali sono gli edifici con percorsi più vicini alla sua posizione	N.I	R0F6

Test	Descrizione	Stato	Requisito
TS0F7	Viene verificato che il sistema permetta all'utente di modificare le proprie credenziali d'accesso	N.I	R0F7
TS0P1	Viene verificato che in ogni schermata il tempo di latenza per ottenere una risposta dal server sia minore di 5 secondi, a meno che non vi siano problemi di connessione	N.I	R0P1
TS0P2	Viene verificato che il tempo di latenza per cambiare la schermata nell'app sia minore di 0.5 secondi, a meno che non sia richiesta l'interazione con il server	N.I	R0P2
TS0Q1	Viene verificato che il codice rispetti le norme e le metriche delle “ <i>Norme di Progetto v4.0.0</i> ” e della sezione Metriche per il codice	N.I	R0Q1
TS0Q2	Viene verificato che i documenti rispettino le norme e le metriche delle “ <i>Norme di Progetto v4.0.0</i> ” e della Metriche per il codice	N.I	R0Q2
TS0Q3	Viene verificato che venga fornito il manuale per l'utente	N.I	R0Q3
TS2F8	Viene verificato che il sistema permetta all'utente di visualizzare un tutorial introduttivo al primo utilizzo dell'app	N.I	R2F8

Tabella 21: Riepilogo test di sistema

B.5 Test di integrazione

I test di integrazione servono per verificare che tutti i diversi componenti del sistema comunichino correttamente tra loro, e che vi sia all'interno del software il flusso di dati atteso.

Verrà utilizzata una strategia di integrazione incrementale per poter sviluppare e verificare più componenti in parallelo. Questo metodo permette di dare priorità ai test relativi alle componenti che vengono ritenute più importanti e quindi sarà possibile partire dalle componenti che soddisfanno i requisiti obbligatori fino ad integrarle con quelle che soddisfano i requisiti opzionali. Inoltre permette di restringere la ricerca dell'errore in caso di test fallito, perché molto probabilmente l'errore si trova nel nuovo componente o dalla sua interazione con il sistema corrente. Non si dovrà escludere il caso in cui il test fallisca perché la nuova istanza di test utilizza un campione di input non trattato in precedenza, portando così il sistema a generare l'errore.

L'integrazione delle parti è bottom-up: innanzitutto verranno inserite le componenti con meno dipendenze funzionali e più funzionalità, cioè che corrispondono ai requisiti obbligatori. Di conseguenza queste componenti saranno testate molte volte in modo da ridurre la possibilità che il prodotto finale contenga difetti, e così si otterrà una versione funzionante nel minor tempo possibile. In seguito si risalirà l'albero delle dipendenze fino all'inserimento delle componenti di alto livello.

Questo metodo è più oneroso rispetto ad altri in quanto richiede che venga generato del codice di supporto, sotto forma di driver e stub, che simuli le componenti mancanti, però permette una maggiore copertura perché testa ripetutamente le componenti più importanti.

Test	Descrizione
TIsavedresults	Verifica che il sistema gestisca correttamente la visualizzazione e il salvataggio dei risultati dei percorsi
TIdatamanager	Verifica che il sistema gestisca correttamente le richieste dei dati salvati in locale
TIpathprogress	Verifica che il sistema gestisca correttamente i dati del percorso selezionato e i risultati delle prove giocate

Test	Descrizione
TIdataserver	Verifica che il sistema gestisca correttamente la modellazione dei dati locali del server
TIgames	Verifica che il sistema gestisca correttamente il percorso e le prove mentre si sta affrontando un percorso
TIserver	Verifica che il sistema gestisca correttamente la ricezione, l'elaborazione e la risposta alle richieste del client
TIlocation	Verifica che il sistema gestisca correttamente l'individuazione e la lettura dei beacon
TIclient	Verifica che il sistema gestisca ed integri correttamente tutte le parti del client, tra cui l'autenticazione, i percorsi e i servizi vari
TICLIPS	Verifica che il client e il server comunichino correttamente tra loro
TIauthentication	Verifica che il sistema gestisca correttamente la registrazione dell'account, il login e la modifica delle proprie credenziali
TIurlrequest	Verifica che il sistema gestisca correttamente le richieste da inviare al server
TIviewController	Verifica che il sistema gestisca correttamente le pagine dell'applicazione e le risposte agli input dell'utente
TIutility	Verifica che il sistema gestisca correttamente le view generali dell'app
TIbuilding	Verifica che il sistema gestisca correttamente le view e le interazioni dell'utente riguardanti gli edifici abilitati e i relativi percorsi
TIurlrequesthandler	Verifica che il sistema gestisca correttamente le richieste del client e le risposte da inviare
TIdata	Verifica che il sistema gestisca correttamente la modellazione dei dati in locale

Tabella 22: Tabella dei test d'integrazione

B.6 Test di unità

Test	Descrizione	Stato	Esito
TU1	Si verifica che la richiesta al server di ricevere i dati dell'applicazione restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU2	Si verifica che la richiesta al server di ricevere i risultati dei percorsi effettuati dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU3	Si verifica che la richiesta al server di salvare i dati del percorso appena effettuato dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU4	Si verifica che la richiesta al server di ricevere i dati del percorso selezionato dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU5	Si verifica che la richiesta al server di ricevere i dati degli edifici più vicini all'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU6	Si verifica che la richiesta al server di effettuare la registrazione del profilo dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU7	Si verifica che la richiesta al server di effettuare il login dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU8	Si verifica che la richiesta al server di effettuare il logout dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU9	Si verifica che la richiesta al server di effettuare la modifica dei dati del profilo dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU10	Si verifica che la richiesta al server di effettuare il controllo sui dati inseriti dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU11	Si verifica che la richiesta al server di ricevere i dati del profilo dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU12	Si verifica che la richiesta al server di cambiare password quando l'utente si è dimenticato la sua restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU13	Si verifica che la richiesta al server di ricevere i dati sulla classifica del percorso selezionato restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU14	Si verifica che la richiesta del programma di ottenere i dati dell'applicazione restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato

Test	Descrizione	Stato	Esito
TU15	Si verifica che la richiesta del programma di ottenere i risultati dei percorsi effettuati dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU16	Si verifica che la richiesta del programma di salvare i dati del percorso appena effettuato dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU17	Si verifica che la richiesta del programma di ottenere i dati del percorso selezionato dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU18	Si verifica che la richiesta del programma di ottenere i dati degli edifici più vicini all'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU19	Si verifica che la richiesta del programma di effettuare la registrazione del profilo dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU20	Si verifica che la richiesta del programma di effettuare il login dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU21	Si verifica che la richiesta del programma di effettuare il logout dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU22	Si verifica che la richiesta del programma di modificare i dati del profilo dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU23	Si verifica che la richiesta del programma di controllare i dati inseriti dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU24	Si verifica che la richiesta del programma di ottenere i dati del profilo dell'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU25	Si verifica che la richiesta del programma di cambiare la password dell'utente quando si è dimenticata la sua restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU26	Si verifica che la richiesta del programma di ottenere i dati della classifica del percorso selezionato dall'utente restituisca, quando è corretta, la risposta giusta, o, quando è errata, l'errore corretto	IMPL.	Superato
TU27	Si verifica che la richiesta del programma di ottenere i dati del percorso, degli edifici, dei risultati dell'utente e dei dati sull'applicazione quando non c'è una connessione Internet attiva restituisca i dati salvati in locale	IMPL.	Superato
TU28	Si verifica che l'errore restituito in caso di pagina errata o di token errato inviato venga gestito correttamente dall'applicazione	IMPL.	Superato

Test	Descrizione	Stato	Esito
TU29	Si verifica che la richiesta di effettuare un'operazione che necessita del GPS o della connessione Internet attiva gestisca correttamente l'errore restituito quando questi servizi non sono attivi	IMPL.	Superato
TU30	Si verifica che le funzioni di lettura e aggiornamento dei dati relativi agli edifici più vicini funzionino correttamente	N.I.	Superato
TU31	Si verifica che le funzioni di lettura e aggiornamento dei dati relativi al percorso selezionato funzionino correttamente	N.I.	Superato
TU32	Si verifica che le funzioni di lettura e aggiornamento dei dati relativi ai risultati dell'utente funzionino correttamente	N.I.	Superato
TU33	Si verifica che i metodi della classe PathProgress funzionino correttamente	IMPL.	Superato
TU34	Si verifica che i metodi delle classi MultipleChoiceTextQuiz, TrueFalseTextQuiz, Path, ProofResult, LinearScoringAlgorithm e Building funzionino correttamente	IMPL.	Superato
TU35	Si verifica che la ricerca della posizione dell'utente tramite il GPS funzioni correttamente	IMPL.	Superato
TU36	Si verifica che i metodi della classe Utility funzionino correttamente	IMPL.	Superato
TU37	Si verifica che il bottone possa essere cliccato e porti all'azione desiderata	N.I.	Superato
TU38	Si verifica che vengano scritti in modo corretto i campi nell'interfaccia	N.I.	Superato
TU39	Si verifica che venga calcolato in modo giusto il punteggio totale	N.I.	Superato
TU40	Si verifica che venga calcolata la distanza da un edificio data una posizione	N.I.	Superato
TU41	Si verifica che venga inserito l'edificio nell'array degli edifici	N.I.	Superato
TU42	Si verifica che venga prodotto l'array di edifici corretti dati latitudine e longitudine	N.I.	Superato
TU43	Si verifica che venga creato il giusto dato di tipo GregorianCalendar a partire dalla stringa data	N.I.	Superato
TU44	Si verifica che il controllo del risultato di una prova sia corretto	N.I.	Superato
TU45	Si verifica che quando viene richiesto il sito web dell'applicazione venga restituita una stringa	IMPL.	Superato
TU46	Si verifica che quando viene richiesta la descrizione dell'applicazione venga restituita una stringa	IMPL.	Superato
TU47	Si verifica che quando viene richiesta la mail di supporto dell'applicazione venga restituita una stringa che sia valida come mail	IMPL.	Superato
TU48	Si verifica che quando viene richiesta l'UUID standard dell'applicazione venga restituita una stringa della lunghezza corretta	IMPL.	Superato
TU49	Si verifica che il sito restituito sia esistente	IMPL.	Superato

Test	Descrizione	Stato	Esito
TU50	Si verifica che quando viene richiesto un edificio senza specificare la latitudine il server ritorni un errore tra 400 e 499	IMPL.	Superato
TU51	Si verifica che quando viene richiesto un edificio senza specificare la longitudine il server ritorni un errore tra 400 e 499	IMPL.	Superato
TU52	Si verifica che quando viene richiesto un edificio senza specificare il numero massimo di risultati il server ritorni un errore tra 400 e 499	IMPL.	Superato
TU53	Si verifica che quando vengono richiesti i building ce ne sia almeno uno	IMPL.	Superato
TU54	si verifica che quando si riceve un building esso contenga tutti i dati e siano delle stringhe	IMPL.	Superato
TU55	si verifica che quando si riceve un building esso contenga i dati di latitudine e longitudine e sia compresi rispettivamente tra -90 e +90 e tra -180 e +180	IMPL.	Superato
TU56	Si verifica che il validatore delle password funzioni in modo corretto	IMPL.	Superato
TU57	Si verifica che la richiesta di registrazione di un nuovo utente senza specificare email, username o password fallisca	IMPL.	Superato
TU58	Si verifica che la richiesta di registrazione di un nuovo utente specificando email, username o password non validi fallisca	IMPL.	Superato
TU59	Si verifica che la richiesta di registrazione di un nuovo utente con dati validi imposti tutti i campi con i dati corretti	IMPL.	Superato
TU60	Si verifica che la richiesta di modifica dei dati utente con dati non validi fallisca	IMPL.	Superato
TU61	Si verifica che la richiesta dei dati utente con dati restituisca i valori precedentemente impostati	IMPL.	Superato
TU62	Si verifica che la richiesta di modifica dei dati utente con dati validi imposti tutti i campi con i dati corretti	IMPL.	Superato
TU63	Si verifica che la richiesta di login con dati mancanti fallisca	IMPL.	Superato
TU64	Si verifica che la richiesta di login con dati validi vada a buon fine	IMPL.	Superato
TU65	Si verifica che i dati dell'utente nel server corrispondano con quelli dell'utente richiesto	IMPL.	Superato
TU66	Si verifica che dopo una richiesta di logout l'utente sia effettivamente non loggato	IMPL.	Superato
TU67	Si verifica che la richiesta di un edificio senza specificare l'ID ritorni un errore del client (400...499)	IMPL.	Superato
TU68	Si verifica che la richiesta di un edificio specificando un ID inesistente ritorni un errore del client (400...499)	IMPL.	Superato
TU69	Si verifica che la richiesta di un edificio specificando l'ID valido ritorni un solo edificio	IMPL.	Superato
TU70	Si verifica che la richiesta di un edificio ritorni tutte le chiavi relative all'edificio, ai suoi, step, ai suoi beacon, alle sue prove	IMPL.	Superato

Test	Descrizione	Stato	Esito
TU71	Si verifica che la richiesta di un edificio ritorni un edificio con lo stesso ID	IMPL.	Superato

Tabella 23: Tabella dei test di Unità

B.7 Tracciamento test di unità - classe

Test	Classe.Metodo
TU1	RequestMaker.getAppInfo
	URLRequest.execute
	URLRequest.objectRequest
	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError
TU5	RequestMaker.getBuildings
	BuildingsRequest.setBody
	URLRequest.execute
	URLRequest.objectRequest
	URLRequest.signalError
	AbstractUrlRequestListener.onResponse
TU6	AbstractUrlRequestListener.onError
	RequestMaker.registration
	RegistrationRequest.setBody
	URLRequest.execute
	URLRequest.objectRequest
	URLRequest.signalError
TU7	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError
	RequestMaker.login
	LoginRequest.setBody
	URLRequest.execute
	URLRequest.objectRequest
TU7	URLRequest.signalError
	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError

Test	Classe.Metodo
TU8	RequestMaker.logout
	URLRequest.execute
	URLRequest.objectRequest
	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError
	LoginManager.sharedManager
	LoginManager.getToken
	LoginManager.isLogged
TU9	RequestMaker.changeProfileData
	ChangeProfileDataRequest.setBody
	URLRequest.execute
	URLRequest.objectRequest
	URLRequest.signalError
	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError
	LoginManager.sharedManager
	LoginManager.getToken
	LoginManager.isLogged
TU10	RequestMaker.check
	CheckRequest.setBody
	URLRequest.execute
	URLRequest.objectRequest
	URLRequest.signalError
	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError
TU11	RequestMaker.getProfileData
	URLRequest.execute
	URLRequest.objectRequest
	AbstractUrlRequestListener.onResponse
	AbstractUrlRequestListener.onError
	LoginManager.sharedManager
	LoginManager.getToken
	LoginManager.isLogged

Test	Classe.Metodo
TU13	DataRequestMaker.getAppInfo
	DataManager.execute
	DataManager.getRemoteData
	DataManager.parseFromUrlRequest
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU17	DataRequestMaker.getBuildings
	DataManager.execute
	DataManager.getRemoteData
	DataManager.parseFromLocal
	DataManager.parseFromUrlRequest
	DataManager.updateLocalData
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU18	LoginManager.registration
	LoginManager.sendResponse
	LoginManager.sendError
	LoginManager.updateLoggedUser
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU19	LoginManager.login
	LoginManager.sendResponse
	LoginManager.sendError
	LoginManager.updateLoggedUser
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU20	LoginManager.logout
	LoginManager.sendResponse
	LoginManager.sendError
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError

Test	Classe.Metodo
TU21	LoginManager.change
	LoginManager.sendResponse
	LoginManager.sendError
	LoginManager.updateLoggedUser
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU22	LoginManager.check
	LoginManager.sendResponse
	LoginManager.sendError
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU23	LoginManager.getProfileData
	LoginManager.sendResponse
	LoginManager.sendError
	AbstractDataManagerListener.onResponse
	AbstractDataManagerListener.onError
TU46	AppInfoProvider.execute
TU47	AppInfoProvider.execute
TU48	AppInfoProvider.execute
TU49	AppInfoProvider.execute
TU50	AppInfoProvider.execute
TU51	BuildingsProvider.execute
TU52	BuildingsProvider.execute
TU53	BuildingsProvider.execute
TU54	BuildingsProvider.execute
TU55	BuildingsProvider.execute
	BuildingsProvider.adPathToBuildings
	BuildingsProvider.addDistance
TU56	BuildingsProvider.execute
TU57	PasswordChecker
TU58	RegistrationHandler.execute
TU59	RegistrationHandler.execute
TU60	RegistrationHandler.execute

Test	Classe.Metodo
TU61	UserDataProvider.execute PostUserData
TU62	UserDataProvider.execute GetUserData
TU63	UserDataProvider.execute PostUserData
TU64	LoginHandler.execute
TU65	LoginHandler.execute
TU66	LoginHandler.execute
TU67	LogoutHandler.execute

Tabella 24: Tabella di tracciamento test di unità - classe

C Esito delle revisioni

Durante lo sviluppo del progetto ci saranno quattro revisioni a cui sottoporsi. Il *committente*_g segnalerà gli errori riscontrati fornendo una valutazione generica dell'andamento del progetto ed una dettagliata per ogni documento. Si elencano di seguito le modifiche apportate in seguito alle revisioni.

C.1 Revisione dei Requisiti

- **Studio di fattibilità:** sono stati corretti gli acronimi scritti in maniera errata.
- **Norme di progetto:** nel documento sono state aggiunte le sezioni che erano state impropriamente inserite nel documento "*Piano di Qualifica v1.0.0*".
- **Analisi dei Requisiti:** sono stati modificati numerosi casi d'uso, cambiando ad esempio la descrizione, lo scenario principale e il padre. Inoltre sono stati aggiunti parecchi casi d'uso, mentre altri sono stati eliminati. Sono stati modificati anche numerosi requisiti sia come conseguenza delle modifiche dei casi d'uso sia per le segnalazioni ricevute. Sono stati aggiunti tanti requisiti e ne sono stati eliminati alcuni.
- **Piano di Progetto:** sono state aggiunte le sezioni sull'Analisi dinamica dei rischi e sono stati cambiati i nomi dei periodi.
- **Piano di Qualifica:** il documento è stato completamente rivisto a seguito delle segnalazioni. Sono state aggiunte altre metriche, soprattutto per la verifica dei processi, ed è stata ampliata l'appendice con i risultati della verifica.

C.2 Revisione di Progettazione

- **Norme di Progetto:** è stata ampliata la sezione "Attività" del "Processo di sviluppo".
- **Analisi dei Requisiti:** sono stati aggiunti alcuni diagrammi delle attività.
- **Specifiche Tecnica:** sono stati aggiunti i vantaggi e gli svantaggi per le tecnologie utilizzate per lo sviluppo dell'applicazione. È stata cancellata la tecnologia *PHP*_g, perché nei casi dove era previsto il suo uso è stato usato *Javascript*_g, inoltre è stato aggiunto il framework *Volley*_g come libreria per l'utilizzo e l'implementazione delle chiamate *REST*_g. È stato deciso di semplificare i diagrammi dividendoli in più parti e descrivendo quindi ogni frammento, permettendo così di focalizzare la presentazione e la spiegazione su poche classi alla volta. È stato sostituito il *design pattern*_g *MVC*_g con il *MVP*_g, dato che è quello più usato per *Android*_g e in più ci permette di conservare l'architettura già progettata, perché di fatto i package individuati sono conformi a questo *pattern*_g architetturale. È stato aggiunto *LoginManager* come Singleton. Sono stati corretti i metodi che risultavano barrati anziché sottolineati. È stata tolta la parola "Server" dai nomi delle classi del server. Aggiunta la spiegazione per quanto riguarda il simbolo "?".
- **Piano di Progetto:** sono stati aggiornati i rischi riscontrati, eliminando quelli decaduti e aggiungendo quelli nuovi.
- **Piano di Qualifica:** è stato aggiunto il modello *CMM*_g come metrica per il controllo di qualità dei processi, inoltre sono stati aggiunti e modificati i test di validazione.
- **Glossario:** è stato aggiunto l'indice.

C.3 Revisione di Qualifica

- **Norme di Progetto:** è stata aggiornata la sezione relativa agli strumenti per il calcolo delle metriche.
- **Analisi dei Requisiti:** sono stati corretti i diagrammi delle attività errati.
- **Specifica Tecnica:** le parti relative al MVC_g sono state sostituite con quelle del MVP_g ; è stata poi modificata l'immagine del Singleton e infine sono state cambiate le descrizioni relative all'architettura.
- **Definizione di Prodotto:** sono stati modificati l'organizzazione delle classi e gli header delle pagine; sono stati aggiunti il tracciamento requisito-classe, quello classe-requisito e la specifica relativa al JSONObject; è stata ampliata qualche descrizione.
- **Manuale Utente:** sono state corrette e ampliate le sezioni relative all'installazione e aggiornamento, alla fase di gioco, alla visualizzazione dei risultati del percorso, alla visualizzazione della classifica del percorso e al cambio delle credenziali.
- **Piano di Progetto:** è stata modificata la sezione relativa ai rischi, facendone decadere alcuni e cambiando la pericolosità e la possibilità che accadano ad altri.
- **Piano di Qualifica:** è stata spostata la [Sezione 5](#) nell'appendice.