



MaaP: MongoDB as an admin Platform

Piano di Qualifica

Versione	4.2.0
Data creazione	2013-11-28
Data ultima modifica	2014-06-26
Stato del Documento	Formale
Uso del Documento	Esterno
Redazione	Alessandro Benetti, Giacomo Pinato, Fabio Miotto,
Verifica	Alberto Garbui, Michele Maso
Approvazione	Mattia Sorgato
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin CoffeeStrap

Sommario

Questo documento ha lo scopo di presentare le strategie adottate dal gruppo Aperture Software nell'ottica del miglioramento continuo e assicurazione della qualità.

Diario delle modifiche

Versione	Data	Autore	Modifiche effettuate
4.2.0	2014-06-26	Mattia Sorgato (RE)	Approvazione documento.
4.1.1	2014-06-26	Michele Maso (VR)	Verifica documento.
4.1.0	2014-06-25	Alberto Garbui (VR)	Verifica documento.
4.0.5	2014-06-24	Giacomo Pinato (VR)	Resoconto attività di Verifica
4.0.4	2014-05-30	Fabio Miotto (VR)	Stesura esiti test
4.0.3	2014-05-16	Alessandro Benetti (PR)	Aggiunti test di unità lato client
4.0.2	2014-05-13	Alessandro Benetti (PR)	Aggiunti test di unità lato server
4.0.1	2014-04-08	Fabio Miotto (AN)	Effettuate correzioni segnalate dal Committente.
3.2.0	2014-03-25	Fabio Miotto (RE)	Approvazione documento.
3.1.1	2014-03-24	Alessandro Benetti (VR)	Verifica documento.
3.1.0	2014-03-21	Fabio Miotto (VR)	Verifica documento.
3.0.8	2014-03-19	Mattia Sorgato (VR)	Resoconto attività di verifica.
3.0.7	2014-03-19	Giacomo Pinato (VR)	Resoconto attività PDCA.
3.0.6	2014-03-12	Giacomo Pinato (PR)	Aggiunta test di validazione e tracciamento.
3.0.5	2014-03-12	Mattia Sorgato (PR)	Aggiunta test di integrazione e tracciamento.
3.0.4	2014-03-10	Mattia Sorgato (PR)	Aggiunta test di sistema.
3.0.3	2014-02-28	Michele Maso (PR)	Incremento documento con progettazione test.
3.0.2	2014-01-17	Alberto Garbui (AM)	Aggiunta analisi RR documento.
3.0.1	2014-01-16	Fabio Miotto (AM)	Aggiunta analisi RR documento.
3.0.1	2014-01-14	Fabio Miotto (AN)	Effettuate correzioni segnalate dal Committente.
2.2.0	2014-01-07	Alberto Garbui (RE)	Approvazione documento.
2.1.0	2014-01-06	Fabio Miotto (VR)	Verifica documento.
2.0.3	2014-01-04	Andrea Perin (VR)	Esito metriche processi.
2.0.2	2014-01-04	Andrea Perin (VR)	Esito metriche documenti.
2.0.1	2014-01-03	Michele Maso (AN)	Incremento documento.
1.2.0	2013-12-16	Giacomo Pinato (RE)	Approvazione documento
1.1.1	2013-12-16	Alessandro Benetti (VR)	Verifica documento
1.1.0	2013-12-15	Fabio Miotto (VR)	Verifica documento
1.0.6	2013-12-13	Giacomo Pinato (RE)	Aggiunto resoconto attività di Verifica e standard di qualità
1.0.4	2013-12-04	Mattia Sorgato (AM)	Aggiunta metriche
1.0.3	2013-12-03	Alberto Garbui (AN)	Aggiunta analisi
1.0.2	2013-12-01	Andrea Perin (RE)	Aggiunta strategie di Verifica

1.0.1	2013-11-28	Andrea Perin (RE)	Creazione documento
-------	------------	-------------------	---------------------

Tabella 1: Registro delle modifiche

Indice

1	Introduzione	8
1.1	Scopo del documento	8
1.2	Scopo del prodotto	8
1.3	Glossario	8
1.4	Riferimenti	8
1.4.1	Normativi	8
1.4.2	Informativi	8
2	Obiettivi di qualità	10
2.1	Qualità di processo	10
2.1.1	Obiettivo	10
2.2	Qualità di prodotto	10
2.2.1	Obiettivi qualità software	11
2.2.2	Obiettivi qualità documentazione	12
3	Strategie	13
3.1	Prodotto Funzionale	13
3.2	Prodotto Affidabile	13
3.3	Prodotto Usabile	13
3.4	Prodotto Manutenibile	13
3.5	Prodotto Portabile	13
3.6	Testo leggibile	13
3.7	Strategie aggiuntive	13
3.7.1	Annidamento accettabile	14
3.7.2	Copertura del codice accettabile	14
4	Misure e metriche	15
4.1	Processi	15
4.1.1	Schedule Variance	15
4.1.2	Budget Variance	15
4.2	Software	15
4.2.1	Soddisfamento requisiti funzionali obbligatori e requisiti obbligatori	15
4.2.2	Complessità ciclomatica	15
4.2.3	Linee di codice per metodo	16
4.2.4	Numero di esecuzioni andate a buon fine	16
4.2.5	Lunghezza manuale utente	16
4.2.6	Accoppiamento afferente	16
4.2.7	Accoppiamento efferente	16
4.2.8	Instabilità	17
4.2.9	Parametri per metodo	17
4.3	Metriche aggiuntive	17
4.3.1	Livelli di annidamento	17
4.3.2	Linee di codice per linee di commento	17
4.3.3	Copertura del codice	18
4.4	Documentazione	18
4.4.1	Indice Gulpease	18
5	Procedure di controllo di qualità	19
5.1	Procedure di controllo di qualità di processo	19
5.2	Procedure di controllo di qualità di prodotto	19

A	Standard di qualità	20
A.1	Standard ISO/IEC 15504	20
A.2	Ciclo di Deming (ciclo PDCA)	21
A.3	Standard ISO/IEC 9126	22
B	Metodologie di verifica e tecniche di analisi	26
B.1	Organizzazione	26
B.2	Pianificazione strategica e temporale	26
B.3	Responsabilità	26
B.4	Risorse	27
B.5	Tecniche di analisi	27
B.5.1	Analisi statica	27
	B.5.1.1 Walkthrough	27
	B.5.1.2 Inspection	27
B.5.2	Analisi dinamica	27
C	Pianificazione dei test	29
C.1	Test di unità	29
C.1.1	Descrizione dei test di unità	29
C.1.2	Tracciamento	40
	C.1.2.1 Tracciamento componente server-test di unità	40
	C.1.2.2 Tracciamento componente client-test di unità	44
C.2	Test di sistema	45
C.2.1	Descrizione dei test di sistema	45
C.3	Test d'integrazione	48
C.3.1	Diagramma d'integrazione	48
C.3.2	Descrizione dei test d'integrazione	49
C.3.3	Tracciamento	51
	C.3.3.1 Tracciamento componente-test d'integrazione	51
	C.3.3.2 Tracciamento test d'integrazione-componente	52
C.4	Test di validazione	53
C.4.1	TV 1 - Lato Utente Business	53
C.4.2	TV 2 - Lato Utente Business	53
C.4.3	TV 3 - Lato Utente Business Amministratore	54
C.4.4	TV 4 - Lato Utente Business Amministratore	55
C.4.5	TV 5 - Lato Utente Business Amministratore	55
C.4.6	TV 6 - Lato Utente Sviluppatore	56
C.4.7	TV 7 - Lato Utente Sviluppatore	56
C.4.8	TV 8 - Lato Utente Sviluppatore	57
C.4.9	Tracciamento	57
	C.4.9.1 Tracciamento Test di Validazione - Requisiti	57
	C.4.9.2 Tracciamento Requisiti - Test di Validazione	59
D	Gestione amministrativa della revisione	63
D.1	Comunicazione e risoluzione di anomalie	63
D.2	Trattamento delle discrepanze	63
E	Resoconto delle attività di verifica	64
E.1	Resoconto PDCA	64
E.2	Riassunto delle attività di verifica	64
E.2.1	Revisione dei Requisiti	64
E.2.2	Revisione di Progettazione	64
E.2.3	Revisione di Qualifica	65
E.3	Dettaglio delle verifiche tramite analisi	65

E.3.1	Analisi dei Requisiti	65
	E.3.1.1 Processi	65
	E.3.1.2 Conclusioni	65
	E.3.1.3 Documenti	66
E.3.2	Analisi in Dettaglio	66
	E.3.2.1 Processi	66
	E.3.2.2 Conclusioni	66
	E.3.2.3 Documenti	66
E.3.3	Progettazione Architetturale	67
	E.3.3.1 Processi	67
	E.3.3.2 Conclusioni	67
	E.3.3.3 Documenti	67
	E.3.3.4 Progettazione	67
E.3.4	Progettazione di dettaglio e codifica	68
	E.3.4.1 Processi	68
	E.3.4.2 Conclusioni	69
	E.3.4.3 Documenti	69
E.4	Dettaglio dell'esito delle revisioni	69
E.4.1	Revisione dei Requisiti	70
E.4.2	Revisione di Progettazione	70

Elenco delle tabelle

1	Registro delle modifiche	3
2	Test di unità parte client	31
3	Test di unità dei servizi del client	32
4	Test di unità parte server	40
5	Tracciamento componente-test unità parte server	44
6	Tracciamento componente-test unità parte client	45
7	Test di sistema	47
8	Test d'integrazione	51
9	Tracciamento componente-test d'integrazione	52
10	Tracciamento test d'integrazione-componente	53
11	Tracciamento Test di validazione - Requisiti	59
12	Tracciamento Requisiti - Test di validazione	62
13	BV e SV calcolati sui documenti durante l'Analisi	65
14	Esiti dell'indice di Gulpease calcolato sui documenti durante l'Analisi	66
15	BV e SV calcolati sui documenti durante l'Analisi in Dettaglio	66
16	Esiti dell'indice di Gulpease calcolato sui documenti durante l'Analisi in Dettaglio	67
17	BV e SV calcolati sui documenti durante la Progettazione Architettuale	67
18	Esiti dell'indice di Gulpease calcolato sui documenti durante la Progettazione	67
19	Tabella accoppiamento componenti	68
20	BV e SV calcolati sui documenti durante la Progettazione di dettaglio e codifica	69
21	Esiti dell'indice di Gulpease calcolato sui documenti durante la Progettazione di dettaglio e codifica	69

Elenco delle figure

1	ISO/IEC 15504	20
2	Ciclo PDCA	22
3	Modello di qualità ISO/IEC 9126	23
4	Diagramma d'integrazione	48

1 Introduzione

1.1 Scopo del documento

Il Piano di Qualifica ha l'obiettivo di definire le strategie adottate dal gruppo Aperture *Software_G* per garantire la qualità del prodotto che verrà sviluppato. Il presente documento descriverà le qualità desiderate che il software dovrà avere, le metriche utilizzate per rendere il prodotto e i processi quantificabili. Per ottenere obiettivi finali qualitativi è necessario un continuo e costante *processo_G* di verifica, per scovare ed eliminare errori in maniera rapida e senza spreco di risorse.

1.2 Scopo del prodotto

Lo scopo del prodotto è produrre un *framework_G* per generare interfacce web di amministrazione dei dati di business basate sullo stack *Node.js_G* e *MongoDB_G*.

L'obiettivo è quello di semplificare il lavoro allo *sviluppatore_G* che dovrà rispondere in modo rapido e standard alle richieste degli esperti di *business_G*.

1.3 Glossario

Al fine di evitare ogni ambiguità nella comprensione del linguaggio utilizzato nel presente documento e, in generale, nella documentazione fornita dal gruppo Aperture Software, ogni termine tecnico, di difficile comprensione o di necessario approfondimento verrà inserito nel documento *Glossario-v4.2.0.pdf*.

Saranno in esso definiti e descritti tutti i termini in corsivo e allo stesso tempo marcati da una lettera "G" maiuscola in pedice nella documentazione fornita.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *Norme-di-progetto-v4.2.0.pdf*;
- **Capitolato d'appalto C1:** MaaP as an admin Platform
<http://www.math.unipd.it/~tullio/IS-1/2013/Progetto/C1.pdf>.

1.4.2 Informativi

- **Piano di Progetto:** *Piano-di-progetto-v4.2.0.pdf*;
- **Glossario:** *Glossario-v4.2.0.pdf*;
- **Slides del corso di Ingegneria del software Modulo A, AA 2013/2014 del prof. Tullio Vardanega:**
<http://www.math.unipd.it/~tullio/IS-1/2013/>;
- **SWEBOK-Version 3 (2004):** capitolo 11-Software Quality
<http://www.computer.org/portal/web/swebok/html/ch11>;
- **Wikipedia:** <http://it.wikipedia.org>;

- **Ian Sommerville, Software Engineering, 9 edizione (2011):**
 - Capitolo 24 - Quality management;
 - Capitolo 26 - Process improvement.
- **Standard ISO/IEC TR 15504 Software process assessment:**
http://en.wikipedia.org/wiki/ISO/IEC_15504;
- **Standard ISO/IEC 9126:2001 Software engineering-product quality:**
http://en.wikipedia.org/wiki/ISO_9126;
- **Budget Variance e Schedule Variance - Dati empirici:**
<http://office.microsoft.com/en-us/project-help/determine-the-right-threshold-for-project-cost.aspx>.
- **Indice Gulpease:**
 - http://it.wikipedia.org/wiki/Indice_Gulpease;
 - <http://xoomer.virgilio.it/roberto-ricci/variabilialeatorie/esperimenti/leggibilita.htm>.
- **Complessità ciclomatica:**
 - http://it.wikipedia.org/wiki/Complessit%C3%A0_ciclomatica.

2 Obiettivi di qualità

In questa sezione verranno mostrati gli obiettivi di qualità che il gruppo si è imposto per la realizzazione del prodotto MaaP: MongoDB as an admin Platform.

Tali obiettivi saranno poi seguiti da:

- **Strategia:** procedimento tramite il quale si raggiunge l'obiettivo prefissato. L'elenco completo delle strategie e la descrizione di ciascuna, sarà possibile consultarlo in sezione 3 di questo documento.
- **Metrica o Misura:** ove possibile, sarà specificata una metrica o una misura per rendere l'obiettivo quantificabile; successivamente in sezione 4 di questo documento sarà fornita una descrizione di ogni metrica e di ogni misura, specificando anche, dove possibile:
 - **Obiettivo Sufficiente:** range stabilito come minimo accettabile, sotto il quale ogni unità o documento non verrà accettato come completo;
 - **Obiettivo Ottimale:** range consigliato e da usare come riferimento, dal quale è necessario scostarsi il meno possibile.

2.1 Qualità di processo

Per garantire la qualità del prodotto finale è necessario migliorare la metodologia che porta alla qualità dei processi che compongono il prodotto. Per fare questo si è deciso di utilizzare lo standard ISO/IEC 15504¹ denominato SPICE². Per applicare il modello appena citato si deve utilizzare il ciclo di Deming³ che ha come obiettivo il miglioramento continuo dei processi nel loro *ciclo di vita_G*.

2.1.1 Obiettivo

Si è deciso di fissare come obiettivo il raggiungimento del terzo livello dello standard sopracitato, ovvero established. Per cercare di ottenere questo, bisognerà:

- Pianificare dettagliatamente i processi utilizzando standard di riferimento;
- Pianificare le risorse da utilizzare.

Sono state scelti due indici che si basano sui costi e i tempi spesi per un processo, per renderlo quantificabile. Tali indici sono:

- **Schedule Variance(SV):** vedi sezione 4.1.1 di questo documento;
- **Budget Variance(BV):** vedi sezione 4.1.2 di questo documento;

Data la scarsa esperienza di cooperazione e di pianificazione delle attività del gruppo, gli obiettivi di questi indici fanno riferimento a convenzioni comuni.

2.2 Qualità di prodotto

La qualità di prodotto può essere suddivisa in due categorie. La prima riguarda il software, mentre la seconda riguarda la documentazione. Per ciascuna categoria il gruppo ha individuato degli obiettivi di qualità da raggiungere, illustrati di seguito.

¹vedi *Appendice_G*, sezione A.1

²Software Process Improvement and Capability Determination

³vedi *appendice*, sezione A.2

2.2.1 Obiettivi qualità software

Per cercare di realizzare e progettare un prodotto software, in accordo con specifiche e standard definiti, ed essere privo di non conformità o difetti, è necessario usare lo standard ISO/IEC 9126⁴, il quale redige e descrive obiettivi qualitativi e fornisce delle linee guida d'utilizzo di metriche al fine di tracciare il progresso nel miglioramento continuo di processo e prodotto. Di seguito verranno elencati gli obiettivi software, basandosi sulle caratteristiche dello standard sopracitato.

- **Funzionalità:** il prodotto software deve soddisfare tutti i requisiti funzionali obbligatori, tutti i requisiti obbligatori trovati e di conseguenza fornire le funzionalità attese;
 - **Strategia adottata:** Prodotto Funzionale(vedi sezione 3.1 di questo documento).
 - **Misura o metrica utilizzata:** Soddisfacimento requisiti funzionali obbligatori e requisiti obbligatori(vedi sezione 4.2.1 di questo documento);
- **Affidabilità:** il prodotto software dovrà essere in grado di funzionare anche in condizioni non ottimali; ad esempio non devono essere prodotti risultati non corretti e la gestione degli errori deve essere trattata in maniera ottimale; verrà misurata la quantità di esecuzioni andate a buon fine.
 - **Strategia adottata:** Prodotto Affidabile(vedi sezione 3.2 di questo documento).
 - **Misura o metrica utilizzata:**
 - * Complessità Ciclomatica(vedi sezione 4.2.2 di questo documento);
 - * Linee di codice per metodo(vedi sezione 4.2.3 di questo documento);
 - * Numero di esecuzioni andate a buon fine(vedi sezione 4.2.5 di questo documento).
- **Usabilità:** essendo legata al soggetto utente, non è facilmente verificabile e misurabile in modo meccanico. Il prodotto dovrà essere comprensibile e facilmente apprendibile;
 - **Strategia adottata:** Prodotto Usabile(vedi sezione 3.4 di questo documento).
 - **Misura o metrica utilizzata:**
 - * Lunghezza manuale utente(vedi sezione 4.2.6 di questo documento);
- **Manutenibilità:** deriva da una buona progettazione architetturale, ma soprattutto da un basso accoppiamento delle componenti;
 - **Strategia adottata:** Prodotto Manutenibile (vedi sezione 3.5 di questo documento).
 - **Misura o metrica utilizzata:**
 - * Accoppiamento efferente ed afferente(vedi sezione 4.2.7 e 4.2.8 di questo documento);
 - * Complessità ciclomatica(vedi sezione 4.2.2 di questo documento);
 - * Lunghezza manuale utente(vedi sezione 4.2.6 di questo documento);
 - * Parametri per metodo(vedi sezione 4.2.10 di questo documento);
 - * Linee di codice per metodo(vedi sezione 4.2.3 di questo documento).
- **Portabilità:** il prodotto software deve essere portabile sotto i vincoli imposti nell'Analisi dei Requisiti
 - **Strategia adottata:** Prodotto Portabile(vedi sezione 3.6 di questo documento).
 - **Misura o metrica utilizzata:**
 - * Parametri per metodo(vedi sezione 4.2.10 di questo documento);
 - * Linee di codice per metodo(vedi sezione 4.2.3 di questo documento).

⁴vedi appendice, sezione A.3

2.2.2 Obiettivi qualità documentazione

Il gruppo Aperture Software si è imposto di produrre documentazione di qualità, nonostante la scarsa inesperienza nel produrre documenti di questo tipo. Proprio per questo motivo è stato introdotto il seguente obiettivo:

- **Leggibilità:** si cercherà di produrre testo caratterizzato da una buona leggibilità.
 - **Strategia adottata:** Testo leggibile(vedi sezione 3.6 di questo documento).
 - **Misura o metrica utilizzata:** Indice di Gulpease(vedi sezione 4.4.1 di questo documento).

3 Strategie

Verranno di seguito riportate le strategie che il gruppo adotta per il raggiungimento degli obiettivi. Alcune strategie fanno riferimento agli obiettivi riguardanti le metriche individuate, mentre altre strategie fanno riferimento agli obiettivi non quantificabili. Per quanto riguarda le strategie di verifica della qualità del prodotto, si utilizzerà l'analisi statica e l'analisi dinamica. Entrambe sono descritte in Appendice B, sezione B.5.

3.1 Prodotto Funzionale

Per quanto riguarda la funzionalità del prodotto, verrà utilizzata la tabella di tracciamento, presente nella sezione 5 del documento Analisi dei Requisiti, che mostra tutti i requisiti obbligatori funzionali e i requisiti obbligatori. Per il raggiungimento di questo obiettivo il prodotto finale dovrà soddisfare tutti i requisiti mostrati in tabella.

3.2 Prodotto Affidabile

Per l'affidabilità del prodotto, si cercherà di scrivere del codice con il minor numero possibile di rami logici. Se questo non risulta possibile, bisogna dividere in più metodi. Prima di effettuare un commit bisogna assicurarsi che quanto scritto fino a quel momento rientri nei range stabiliti delle metriche utilizzate.

3.3 Prodotto Usabile

Per quanto riguarda l'usabilità del prodotto, si cercherà di realizzarlo pensando di farlo il più semplice possibile, ponendo maggiore attenzione all'interazione che esso avrà con chi lo usa. Inoltre il manuale utente che verrà rilasciato insieme al prodotto dovrà essere il più semplice possibile e di facile comprensione; esso dovrà avere un buon valore di indice di leggibilità.

3.4 Prodotto Manutenibile

Per quanto riguarda la manutenibilità del prodotto, ogni volta che verranno progettate delle componenti si procederà al calcolo dell'accoppiamento afferente ed efferente; dopodiché si procede al calcolo dell'instabilità cercando di rientrare nel valore ottimale corrispondente. Inoltre prima di effettuare un commit, ogni pezzo di codice scritto deve soddisfare le metriche individuate.

3.5 Prodotto Portabile

Per la portabilità del prodotto, verrà utilizzata la tabella di tracciamento dei requisiti di vincolo presente nel documento Analisi dei Requisiti. Per assicurarsi maggior portabilità dovranno essere soddisfatte le tipologie di requisiti indicate.

3.6 Testo leggibile

Per cercare di ottenere un testo leggibile e che soddisfi l'indice a esso associato, il gruppo si impegna a scrivere testo contenente frasi corte ma significative. Inoltre dovranno essere usate parole di uso comune e non troppo lunghe.

3.7 Strategie aggiuntive

Verranno ora elencate le strategie che il gruppo utilizzerà per ottenere più qualità possibile.

3.7.1 Annidamento accettabile

Ogni volta che si realizza un metodo, si controlla se il numero di chiamate annidate rientra nei parametri prestabiliti.

3.7.2 Copertura del codice accettabile

Ogni volta che si realizza un metodo, esso deve essere il più semplice possibile, in modo tale che non necessiti di testing.

4 Misure e metriche

In base agli obiettivi di qualità descritti nella sezione 2 di questo documento, il gruppo ha individuato una serie di metriche e misure che permetteranno di rendere gli obiettivi quantificabili e di conseguenza verificabili. Come descritto sempre in tale sezione, ogni metrica e ogni misura sarà accompagnata, dove possibile, da un range di valori. Esse saranno divise per le categorie Processi, Software e Documentazione.

4.1 Processi

4.1.1 Schedule Variance

Indica se si è in linea, in anticipo o in ritardo rispetto alla schedulazione delle attività di progetto pianificate nella *baseline_G*. È un indicatore di efficacia e se il suo valore è > 0 allora il progetto sta avanzando con maggiore velocità rispetto a quanto pianificato. Viceversa se negativo. Gli obiettivi fissati sono:

- **Obiettivo Sufficiente:** $[\geq -(\text{costo preventivo per fase} * 5\%)]$;
- **Obiettivo Ottimale:** $[\geq 0]$.

4.1.2 Budget Variance

Indica se alla data corrente si è speso di più o di meno rispetto a quanto si era pianificato. Se tale valore è > 0 allora il progetto sta consumando il proprio budget con minor velocità rispetto a quanto pianificato. Viceversa se negativo.

- **Obiettivo Sufficiente:** $[\geq -(\text{costo preventivo per fase} * 10\%)]$;
- **Obiettivo Ottimale:** $[\geq 0]$.

4.2 Software

4.2.1 Soddisfacimento requisiti funzionali obbligatori e requisiti obbligatori

Fornisce un valore che indica quanti requisiti, delle tipologie indicate, sono stati soddisfatti. Degli obiettivi ragionevoli sono:

- **Obiettivo Sufficiente:**[100% soddisfacimento requisiti funzionali obbligatori e requisiti obbligatori]
- **Obiettivo Ottimale:**[100% soddisfacimento di tutti i requisiti individuati]

4.2.2 Complessità ciclomatica

La complessità ciclomatica è una metrica software applicabile singolarmente a funzioni, moduli, metodi e classi di un *programma_G*. Questa metrica è calcolata utilizzando il grafo di controllo di flusso del programma, ovvero i nodi del grafo rappresentano gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo di istruzioni può essere eseguito subito dopo il primo gruppo. Alti valori di questa metrica implicano una scarsa manutenibilità del software, mentre valori troppo bassi possono indicare un'altrettanta bassa efficienza del software. Un modulo, con complessità ciclomatica elevata, necessita di più *testing_G* rispetto ad un altro modulo con complessità ciclomatica minore. La complessità è quindi definita come:

$$v(G) = e - n + 2p$$

dove:

- $v(G)$ = complessità ciclomantica del grafo G ;
- e = numero di archi del grafo;
- n = numero di nodi del grafo;
- p = numero di componenti connesse.

Degli obiettivi ragionevoli per questa metrica sono i seguenti:

- **Obiettivo Sufficiente:** [11–15];
- **Obiettivo Ottimale:** $[1-10]^5$.

4.2.3 Linee di codice per metodo

- **Obiettivo Sufficiente:**[11–30]
- **Obiettivo Ottimale:**[10]

4.2.4 Numero di esecuzioni andate a buon fine

- **Obiettivo Sufficiente:**[98%]
- **Obiettivo Ottimale:**[100%]

4.2.5 Lunghezza manuale utente

Fornisce un valore che indica il numero di pagine che un manuale utente deve avere. I valori per questi obiettivi verranno forniti nella prossima revisione, in quanto non è stato completato il Manuale Utente per questa revisione.

- **Obiettivo Sufficiente:**[]
- **Obiettivo Ottimale:**[]

4.2.6 Accoppiamento afferente

Questo valore indica la quantità di classi esterne ad un *package*_{*G*} che dipendono da classi interne allo stesso. Un alto valore di accoppiamento in una singola classe del package influisce sull'accoppiamento dell'intero package. Questo fatto non è necessariamente un errore di progettazione, ma il package in esame può rappresentare un punto critico del software. Per contro, un package con basso fattore di accoppiamento può delineare una scarsa utilità del package stesso, che probabilmente andrebbe inglobato con altri package. Esso è legato, insieme all'accoppiamento efferente, ad una metrica importante, ovvero l'instabilità.

4.2.7 Accoppiamento efferente

Questo fattore indica l'accoppiamento contrario, ovvero il numero di classi interne al package che dipendono da classi esterne. Più questo indice è basso, più indipendente è il package stesso.

⁵Il valore 10 come massimo è stato calcolato da T.J.McCabe, inventore della metrica.

4.2.8 Instabilità

Il fattore di instabilità di un package indica la possibilità di modifica del package senza influire sulla stabilità del software ad esso dipendente. Questo indice è calcolato con la formula seguente:

$$I = Ce / (Ca + Ce)$$

dove Ca è l'accoppiamento afferente e Ce l'accoppiamento efferente. Gli obiettivi forniti da *best practice*⁶ sono i seguenti:

- **Obiettivo Sufficiente:** [0.4–0.8];
- **Obiettivo Ottimale:** [0.0–0.3].

4.2.9 Parametri per metodo

Un alto numero di parametri per metodo denota un'eccessiva complessità del metodo stesso, comportandone probabilmente un'ulteriore lunghezza non accettabile. È buona norma scrivere dei metodi con pochi parametri, al fine di ottenere procedure specifiche e atomiche, di conseguenza facilmente assegnabili e verificabili. Degli obiettivi validi per questa metrica sono:

- **Obiettivo Sufficiente:** [5–8];
- **Obiettivo Ottimale:** [0–4].

4.3 Metriche aggiuntive

Per raffinare ulteriormente la qualità, soprattutto per la manutenibilità e l'affidabilità, vengono utilizzate delle ulteriori metriche.

4.3.1 Livelli di annidamento

Il numero di livelli di annidamento dei metodi rappresenta la quantità di richiami di altri metodi all'interno di uno stesso *metodo_G*. Un elevato livello di annidamento definisce un'elevata complessità del codice e di altrettanta comprensione dello stesso. Gli obiettivi stimati per questa metrica sono:

- **Obiettivo Sufficiente:** [4–6];
- **Obiettivo Ottimale:** [1–3].

4.3.2 Linee di codice per linee di commento

Questo numero indica il rapporto tra linee di codice e linee di commento, per avere un fattore di commenti all'interno di un'unità software. In generale, un alto grado di commento del codice porta ad una maggiore manutenibilità ed informazione per uno *sviluppatore_G*. Gli obiettivi stimati sono:

- **Obiettivo Sufficiente:** [> 0.25];
- **Obiettivo Ottimale:** [0.26–0.30]⁷.

⁶Range ricavati da <http://staff.unak.is/andy/StaticAnalysis0809/metrics/i.html>

⁷Il valore di 0.30 è stato calcolato dal rapporto 22/78, derivato dalle medie di Ohloh <https://www.ohloh.net/p/firefox/factoids#FactoidCommentsLow>

4.3.3 Copertura del codice

Questo fattore indica la percentuale di codice coperto durante l'esecuzione dei test. Più alta sarà la percentuale, minore sarà la possibilità di errori riscontrabili nell'esecuzione del software. Per abbassare questo indice sarà sufficiente scrivere metodi semplici che non necessitino di testing. Il valore ideale di 100% indica che tutte le porzioni di codice sono testate da uno o più test. Gli obiettivi stimati per questa metrica sono:

- **Obiettivo Sufficiente:** [42%–65%];
- **Obiettivo Ottimale:** [66%–100%].

4.4 Documentazione

4.4.1 Indice Gulpease

L'indice Gulpease è un indice di leggibilità del testo che basa il suo calcolo su componenti del testo enumerabili meccanicamente, così da rendere automatico il processo di Verifica. Consente di misurare la complessità dello stile di un documento. L'indice di Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero di lettere. La formula per il suo calcolo è la 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 il valore 100 indica la leggibilità più alta e 0 la leggibilità più bassa. In generale risulta che testi con un indice

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

L'indice prevede un intervallo di valori tra 0 e 100, dove 100 esprime la leggibilità massima. I nostri obiettivi per l'indice Gulpease sono i seguenti:

- **Obiettivo ottimale:** [51–100];
- **Obiettivo sufficiente:** [40–50].

5 Procedure di controllo di qualità

5.1 Procedure di controllo di qualità di processo

Per garantire la qualità dei processi si utilizza il ciclo PDCA⁸. Questo principio permette un continuo miglioramento della qualità di tutti i processi coinvolti nella realizzazione del prodotto finale. Per controllare la qualità bisogna che i processi siano pianificati dettagliatamente, che le risorse siano individuate e ripartite in maniera quantificabile e che ci sia un controllo sui processi. Lo sviluppo di quanto scritto prima è descritto dettagliatamente nel *Piano di Progetto 4.2.0*. Inoltre verrà monitorata la qualità dei processi con l'analisi continua della qualità del prodotto.

5.2 Procedure di controllo di qualità di prodotto

Per garantire il controllo di qualità si utilizza:

- **Quality Assurance:** tradotta in assicurazione di qualità, è l'insieme di processi che hanno come fine il miglioramento e il perseguimento della qualità. L'intenzione di un team di lavoro consiste nell'ottenere quella che si dice *correction by construction*, ovvero correttezza per costruzione;
- **Strategie proattive:** l'insieme delle strategie proattive, le cui procedure sono descritte nel documento *Norme di Progetto*, permettono di garantire qualità a tempo zero, limitando *attività_G* di verifica che hanno un costo non indifferente.
- **Verifica:** è la valutazione che un prodotto, *servizio_G* o sistema, sia conforme a regole, requisiti, specifiche o condizioni imposte. È spesso un processo interno e differisce dalla validazione. In analogia, l'attività di Verifica deve rispondere alla domanda: *did i built the system right?*, ovvero ho costruito il sistema in modo corretto?;
- **Validazione:** è l'assicurazione che un prodotto, servizio o sistema, incontri le necessità che i clienti o gli stakeholder identificano. Spesso comporta l'accettazione e l'idoneità con clienti esterni. In questo caso la domanda è: *did i built the right system?*, tradotto in ho costruito il sistema giusto?.

⁸Alias, Ciclo di Deming, vedi appendice, sezione A.2

A Standard di qualità

A.1 Standard ISO/IEC 15504

Lo standard ISO/IEC 15504, anche conosciuto come SPICE (Software Process Improvement and Capability Determination, ovvero miglioramento di processi software e determinazione di capacità) è un insieme di documenti e di standard tecnici per lo sviluppo software. Questo documento viene utilizzato nel perseguimento della qualità di processo in quanto stabilisce una struttura per la definizione degli obiettivi per il miglioramento dei processi stessi. Lo standard dichiara che ogni processo deve essere sottoposto ad un controllo continuo, ripetibile e quantificabile, al fine di individuare i punti critici che impediscono di raggiungere gli obiettivi prefissati, e misurare i miglioramenti.

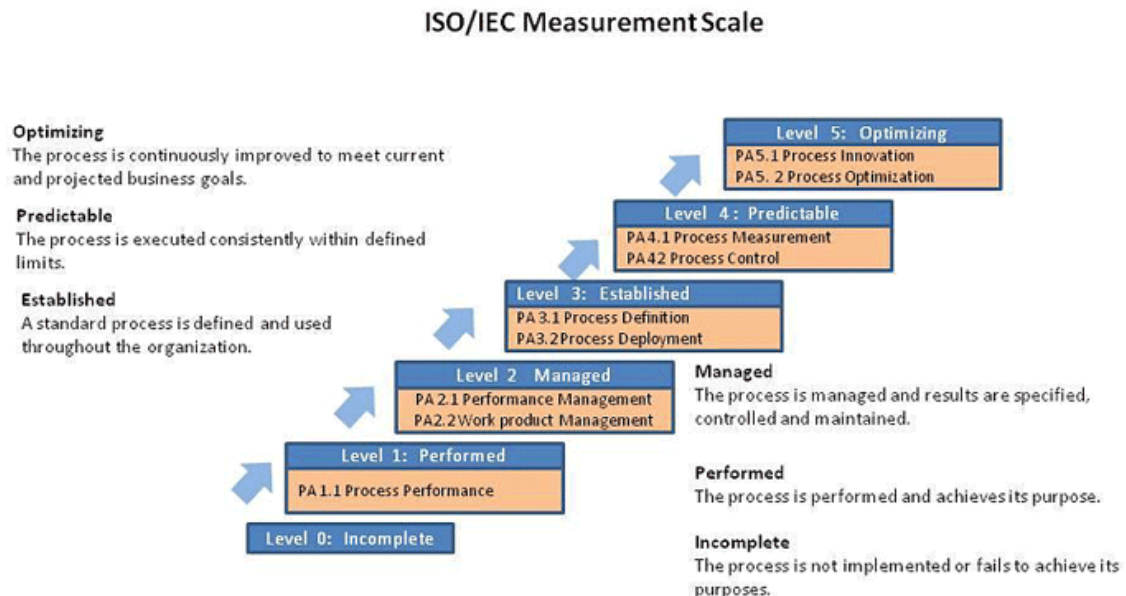


Figura 1: ISO/IEC 15504

Secondo SPICE, un processo può essere classificato in base al suo livello di maturità, in una *scala_G* da 1 a 6, con annessi i livelli di capacità ad ogni livello:

1. **Incomplete:** i risultati del processo non esistono o non sono appropriati;
2. **Performed:** si ottengono dei risultati, ma in un modo non specificato o non prevedibile;
 - **Process Performance:** capacità del processo di produrre degli output da degli *input_G*.
3. **Managed:** l'esecuzione è pianificata e tracciata, il prodotto è conforme a standard e requisiti specifici;

- **Performance Management:** capacità del processo di produrre un output coerente con gli obiettivi del processo;
 - **Work Product Management:** capacità del processo di creare un risultato documentato, controllato e verificato.
4. **Established:** il processo è eseguito e controllato riferendosi a dei buoni principi di ingegneria del software;
- **Process Definition:** il processo fa riferimento a degli standard di processo per definire i risultati attesi;
 - **Process Deployment_G:** capacità del processo di utilizzare risorse appropriate per il raggiungimento degli obiettivi.
5. **Predictable:** il processo è eseguito consistentemente con dei limiti di controllo definiti, per raggiungere altrettanto definiti obiettivi di processo;
- **Process Measurement:** capacità di definizione di obiettivi e metriche di prodotto e di processo, con cui garantire il raggiungimento di obiettivi aziendali;
 - **Process Control:** capacità di controllo tramite metriche di progetto e prodotto definite, per puntare al miglioramento.
6. **Optimizing:** l'esecuzione del processo è ottimizzata per soddisfare bisogni correnti e futuri, e il processo soddisfa ripetibilmente i suoi obiettivi prefissati;
- **Process Innovation:** capacità di gestione di eventuali cambiamenti nel prodotto in modo controllato ed efficace;
 - **Continuous Optimization:** capacità di identificare e applicare modifiche atte al miglioramento dei processi aziendali.

Per finire, lo standard definisce 4 stadi di misurazione degli attributi di un processo, suddivisi in:

- **N**, non adeguato o non posseduto;
- **P**, parzialmente posseduto;
- **L**, largamente posseduto;
- **F**, completamente posseduto.

A.2 Ciclo di Deming (ciclo PDCA)

Il ciclo di Deming o Deming Cycle (ciclo di PDCA - plan-do-check-act) è un modello studiato per il miglioramento continuo della qualità in un'ottica a lungo raggio. Serve per promuovere una cultura della qualità che è tesa al miglioramento continuo dei processi e all'utilizzo ottimale delle risorse. Questo miglioramento è ottenuto tramite la continua esecuzione di quattro fasi fondamentali, descritte nell'elenco sottostante. Per migliorare la qualità e soddisfare il cliente, le quattro fasi devono ruotare costantemente, tenendo come criterio principale la qualità. La sequenza logica dei quattro punti ripetuti per un miglioramento continuo è la seguente:

- **P** - Plan. Pianificazione.
- **D** - Do. Esecuzione del programma, dapprima in contesti circoscritti.
- **C** - Check. Test e controllo, studio e raccolta dei risultati e dei riscontri.

- **A** - Act. Azione per rendere definitivo e/o migliorare il processo.

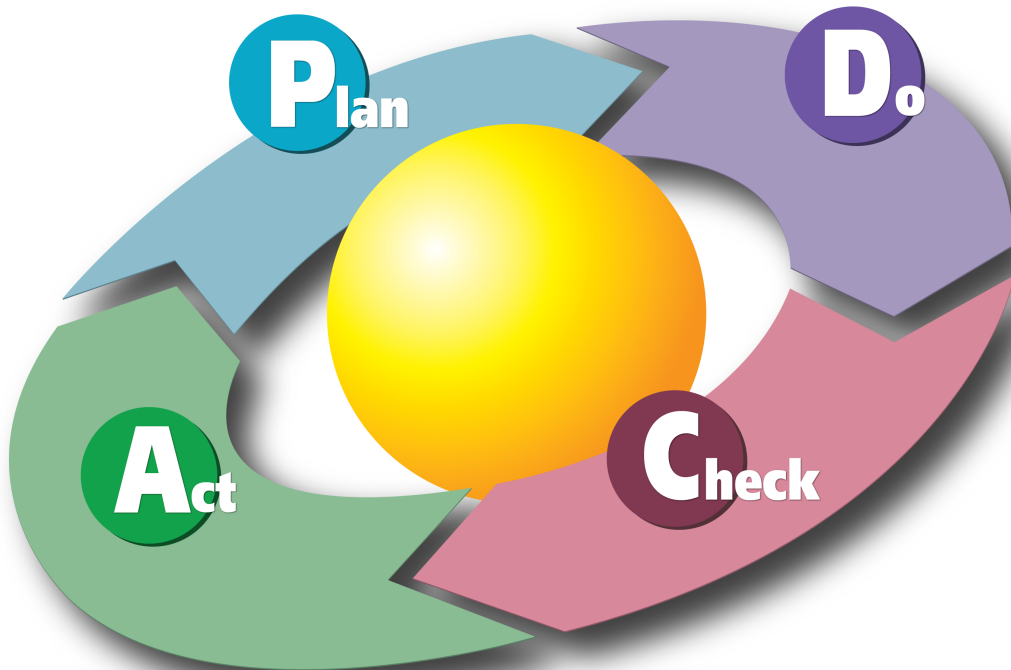


Figura 2: Ciclo PDCA

A.3 Standard ISO/IEC 9126

Con la sigla ISO/IEC 9126 si individua una serie di norme e linee guida, sviluppate dall'*ISO_G* (Organizzazione internazionale per la normazione) in collaborazione con l'*IEC_G* (Commissione Elettrotecnica Internazionale), preposte a descrivere un modello di qualità del software. Il modello propone un approccio alla qualità in modo tale che le società di software possano migliorare l'organizzazione e i processi e, quindi come conseguenza concreta, la qualità del prodotto sviluppato. Ci sono 3 tipi di qualità:

- **Qualità in uso:** le metriche in uso, specificate nella norma ISO/IEC 9126-1, misurano la qualità del prodotto software dal punto di vista dell'utilizzatore, che le usa internamente ad uno specifico sistema e contesto;
- **Qualità esterna:** le metriche esterne, specificate nella norma ISO/IEC 9126-2, misurano i comportamenti del software sulla base dei test, dall'operatività e dall'osservazione durante la sua esecuzione, in funzione degli obiettivi stabiliti in un contesto tecnico rilevante o di *business_G*;

- **Qualità interna:** la qualità interna, più precisamente la metrica interna, è specificata nella norma ISO/IEC 9126-3 e si applica al software non eseguibile (ad esempio il codice sorgente) durante le fasi di Progettazione e Codifica. Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale, poiché gli attributi interni influiscono su quelli esterni e quelli in uso. Le metriche interne permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il software eseguibile. Esistono metriche che possono simulare il comportamento del prodotto finale tramite simulazioni.

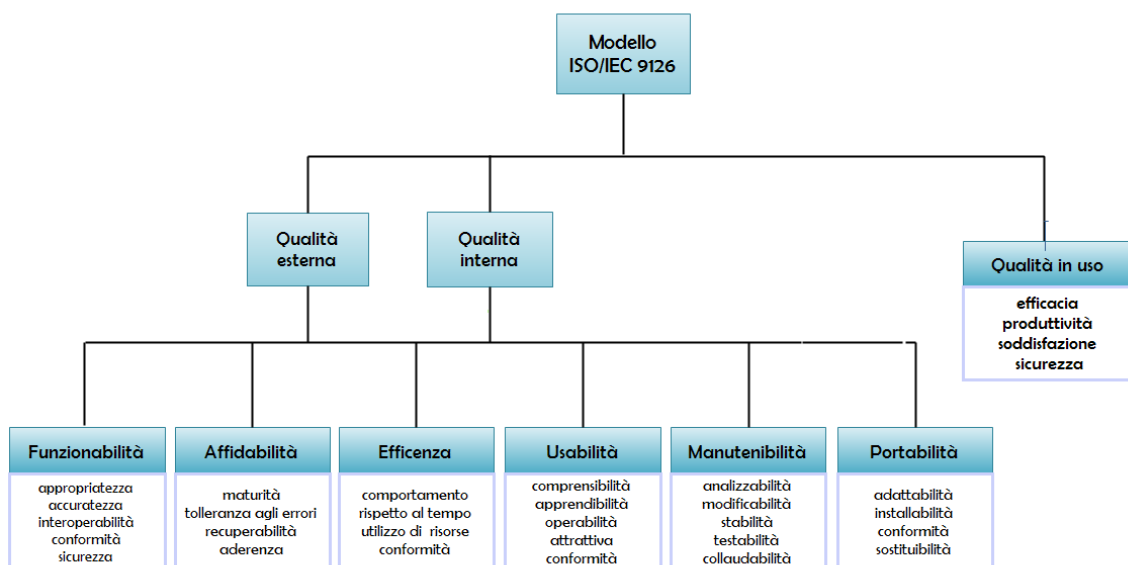


Figura 3: Modello di qualità ISO/IEC 9126

Il presente standard definisce sei caratteristiche di qualità che ogni prodotto software deve perseguire, al fine di garantire la conformità agli standard con efficienza ed efficacia. Le caratteristiche delle qualità in uso esulano dal presente progetto didattico, in quanto non è prevista l'attività di manutenzione conseguente al rilascio del prodotto. Quindi ci soffermiamo all'analisi della qualità interna ed esterna dello standard ISO/IEC 9126. Le caratteristiche che un prodotto deve avere sono le seguenti:

- **Funzionalità:** capacità di un prodotto software di fornire funzioni che soddisfano esigenze stabili, necessarie per operare sotto condizioni specifiche;
 - *Appropriatezza:* rappresenta la capacità del prodotto software di fornire un appropriato insieme di funzioni per gli specificati compiti ed obiettivi prefissati all'*utente_G*;
 - *Accuratezza:* la capacità del prodotto software di fornire i risultati concordati o precisi effetti richiesti;
 - *Interoperabilità:* la capacità del prodotto software di interagire ed operare con uno o più sistemi specificati;

- *Conformità*: la capacità del prodotto software di aderire agli standard, convenzioni e regolamentazioni rilevanti al settore operativo a cui vengono applicati;
- *Sicurezza*: la capacità del prodotto software di proteggere informazioni e i dati negando in ogni modo che persone o sistemi non autorizzati possano accedervi o modificarli, e che a persone o sistemi effettivamente autorizzati non sia negato l'accesso ad essi.
- **Affidabilità**: capacità del prodotto software di mantenere uno specificato livello di prestazioni quando usato in date condizioni per un dato periodo;
 - *Maturità*: capacità di un prodotto software di evitare che si verificano errori, malfunzionamenti o siano prodotti risultati non corretti;
 - *Tolleranza degli errori*: capacità di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto;
 - *Recuperabilità*: capacità di un prodotto di ripristinare il livello appropriato di prestazioni e di recupero delle informazioni rilevanti, in seguito a un malfunzionamento. A seguito di un errore, il software può risultare non accessibile per un determinato periodo di tempo, questo arco di tempo è valutato proprio dalla caratteristica di recuperabilità;
 - *Aderenza*: capacità di aderire a standard, regole e convenzioni inerenti all'affidabilità.
- **Efficienza**: capacità di fornire appropriate prestazioni relativamente alla quantità di risorse usate;
 - *Comportamento rispetto al tempo*: capacità di fornire adeguati tempi di risposta, elaborazione e velocità di attraversamento, sotto determinate condizioni;
 - *Utilizzo delle risorse*: capacità di utilizzo di quantità e tipo di risorse in maniera adeguata;
 - *Conformità*: capacità di aderire a standard e specifiche sull'efficienza.
- **Usabilità**: capacità del prodotto software di essere capito, appreso, usato e benaccetto dall'utente, quando usato sotto condizioni specificate.
 - *Comprensibilità*: esprime la facilità di comprensione dei concetti del prodotto, mettendo in grado l'utente di comprendere se il software è appropriato;
 - *Apprendibilità*: capacità di ridurre l'impegno richiesto agli utenti per imparare ad usare la sua applicazione;
 - *Operabilità*: capacità di mettere in condizione gli utenti di farne uso per i propri scopi e controllarne l'uso;
 - *Attrattiva*: capacità del software di essere piacevole per l'utente che ne fa uso;
 - *Conformità*: capacità del software di aderire a standard o convenzioni relativi all'usabilità.
- **Manutenibilità**: capacità del software di essere modificato, includendo correzioni, miglioramenti o adattamenti;
 - *Analizzabilità*: rappresenta la facilità con la quale è possibile analizzare il codice per localizzare un errore nello stesso;
 - *Modificabilità*: capacità del prodotto software di permettere l'implementazione di una specificata modifica (sostituzioni componenti);
 - *Stabilità*: capacità del software di evitare effetti inaspettati derivanti da modifiche errate;
 - *Testabilità*: capacità di essere facilmente testato per validare le modifiche apportate al software.

- **Portabilità_G**: capacità del software di essere trasportato da un ambiente di lavoro ad un altro. (Ambiente che può variare dall'hardware al sistema operativo);
 - *Adattabilità*: capacità del software di essere adattato per differenti ambienti operativi senza dover applicare modifiche diverse da quelle fornite per il software considerato;
 - *Installabilità*: capacità del software di essere installato in uno specificato ambiente;
 - *Conformità*: capacità del prodotto software di aderire a standard e convenzioni relative alla portabilità;
 - *Sostituibilità*: capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti nello stesso ambiente.

B Metodologie di verifica e tecniche di analisi

B.1 Organizzazione

Per ogni processo attuato ci sono delle attività di Verifica e per ogni processo realizzato viene verificata la qualità del processo stesso e la qualità dell'eventuale prodotto ottenuto da esso. Ogni periodo di tempo antecedente la consegna di revisione descritto nel Piano di Progetto, necessita di attività di Verifica:

- **Analisi:** si seguono i metodi di Verifica descritti nelle Norme di Progetto sui documenti prodotti e i processi attuati. La messa in opera di tali tecniche è descritta nell'appendice sezione E.2.1;
- **Analisi di Dettaglio:** si verificano i processi che determinano l'incremento dei documenti redatti per il precedente periodo di Analisi, e si verificano i prodotti generati dai relativi processi, seguendo le Norme di Progetto;
- **Progettazione Architettuale:** si verificano i processi che determinano l'incremento dei documenti redatti per il precedente periodo di Analisi in Dettaglio e si verificano i prodotti generati dai relativi processi; inoltre si verificano processi e prodotti per l'attività di Progettazione Architettuale, seguendo le Norme di Progetto. La messa in opera di tale attività è descritta nell'appendice sezione E.2.2;
- **Progettazione di Dettaglio e Codifica:** si verificano i processi che determinano l'incremento dei documenti redatti per il precedente periodo di Progettazione Architettuale e si verificano i prodotti generati dai relativi processi; inoltre si verificano processi e prodotti per l'attività di Progettazione di Dettaglio e Codifica, seguendo le Norme di Progetto. La messa in opera di tale attività è descritta nell'appendice sezione E.2.3.

Questa sezione conterrà i riferimenti alle successive attività di Verifica, aggiornati dopo ogni fase di produzione e di $test_G$ del prodotto. La stesura dei documenti è l'attività principale e costante nello svolgimento del progetto, mentre il processo di Verifica viene diviso in due attività. In ogni documento è presente un diario delle modifiche per mantenere una cronologia delle attività svolte e di chi le ha svolte.

B.2 Pianificazione strategica e temporale

Avendo scadenze prefissate nel Piano di Progetto, dobbiamo garantire che le attività di Verifica di tutti i documenti e prodotti debbano essere sistematiche, disciplinate e quantificabili. Procedendo in questa maniera si correggono gli errori il prima possibile. La metodologia da seguire per l'individuazione e correzione degli errori è descritta nelle Norme di Progetto. Ogni attività di redazione di documenti e di scrittura del $codice_G$ è stata preceduta da uno studio iniziale sull'impaginazione dei documenti e del contenuto degli stessi. Questo serve per minimizzare la possibilità di incorrere in errori di tipo concettuale e tecnico.

B.3 Responsabilità

Per garantire che il processo di Verifica sia disciplinato, sistematico e quantificabile, bisogna attribuire responsabilità a specifici ruoli di progetto. I ruoli sono Responsabile di Progetto e Verificatore. I compiti di ciascun ruolo sono descritti nelle *Norme di progetto-v4.2.0.pdf*, rispettivamente nelle sezioni 3.1 e 3.5.

B.4 Risorse

Per raggiungere gli obiettivi di qualità prefissati sono necessarie risorse umane e tecnologiche, suddivise rispettivamente in strumenti software e *hardware_G* utilizzati dai componenti del gruppo per effettuare Verifica su processi e prodotti. I ruoli maggiormente coinvolti nella responsabilità delle attività di Verifica e Validazione sono il Responsabile di Progetto e il Verificatore e i rispettivi compiti sono descritti dettagliatamente nelle Norme di Progetto. Per facilitare il lavoro dei Verificatori sono stati usati degli strumenti automatici che eseguono controlli sistematici sui prodotti generati. Questi strumenti sono descritti nelle Norme di Progetto.

B.5 Tecniche di analisi

Di seguito verranno elencate le tecniche di analisi che il gruppo adotterà durante lo svolgimento del progetto.

B.5.1 Analisi statica

Questa tipologia di analisi può essere applicata sia al codice che alla documentazione, dato che prevede l'utilizzo di tecniche generali per ogni tipo di prodotto del team.

Le tecniche di analisi statica sono:

- **Walkthrough;**
- **Inspection;**

B.5.1.1 Walkthrough Questa tecnica utilizza una scansione ampia e non mirata dell'oggetto in verifica, data la mancanza di esperienza *best practice_G* del Verificatore. L'attuazione di questa tecnica di Analisi è quindi molto onerosa, per questo sarà nostro obiettivo renderla più parallelizzabile possibile, così da ridurre i costi di Verifica e per essere più efficace ed efficiente. Si comincia con una attività preliminare di lettura, seguita da una individuazione degli errori; poi si procede con la correzione degli stessi e con una successiva attività di lettura per controllare le modifiche apportate. Dopo ogni attività di Verifica tramite Walkthrough, sperabilmente avremo trovato la maggior parte degli errori, fornendoci una visione delle erroneità commesse, di conseguenza potremo raffinare l'Analisi e avvicinarci alla metodologia di Inspection.

B.5.1.2 Inspection Questa tecnica è un'evoluzione del Walkthrough e applica una ricerca più mirata e specifica. È possibile utilizzare questa tecnica dopo aver acquisito dimestichezza con l'attività di Verifica, stilando una lista di controllo contenente i maggiori errori riscontrati applicando la tecnica di Walkthrough, quindi non sarà possibile utilizzarla fin da subito in quanto la lista inizialmente è vuota. È obiettivo di una fase di Inspection la ricerca mirata di errori, aumentando l'efficienza della Verifica e riducendo i costi in termini di tempo e risorse. Durante l'utilizzo della tecnica di Inspection la lista verrà aggiornata; la lista risultante è in appendice B delle *Norme_di_progetto_v4.2.0.pdf*.

B.5.2 Analisi dinamica

Questa particolare tipologia di Analisi si applica solamente ai prodotti software sviluppati dal team, mediante l'utilizzo di test progettati e scritti appositamente per verificare la correttezza dei prodotti e la loro effettiva validazione. Di seguito analizzeremo i 5 tipi di test attuati nelle varie parti del progetto:

- **Test di unità:** attività di Verifica svolta su ogni singola unità software del sistema, mediante l'utilizzo di *stub_G*, *driver_G* e *logger_G*. Un'unità è la più piccola parte di lavoro che viene assegnata individualmente al *programmatore_G*, successivamente sarà prodotta e verificata singolarmente. Mediante tali test viene verificato il corretto funzionamento dei moduli di cui il sistema è composto, in modo da cancellare eventuali errori di implementazione commessi dai programmatori;
- **Test di integrazione:** attività di Verifica che controlla la corretta integrazione di più unità software aggiunte in maniera incrementale, il cui scopo è analizzare che la combinazione delle unità software funzioni come attesa. Grazie a questo test si possono rilevare errori non riscontrabili nei test di unità e comportamenti inaspettati di componenti software già esistenti rilasciati da altri fornitori che non interagiscono correttamente tra di loro. Anche in questa attività, per poter simulare le unità nell'integrazione, vengono create unità fittizie specifiche, come stub e driver, in modo da replicare componenti non ancora sviluppate in modo da non falsare i test;
- **Test di sistema:** questo test si propone di validare il prodotto, una volta che si è stabilita la sua *versione_G* definitiva. Questo test verifica che la copertura dei requisiti obbligatori decisi nel periodo di tempo dedicato all'Analisi in Dettaglio sia completa;
- **Test di regressione:** attività di Verifica che consiste nel ripetere i test già effettuati su una *componente_G*, ogni qualvolta quella componente venga modificata o aggiornata; un aiuto lo fornisce il *tracciamento_G* delle componenti che permette di scovare e ripetere in modo semplificato i test di unità, di regressione e possibilmente quelli di sistema che sono stati potenzialmente alterati dalla modifica;
- **Test di accettazione:** test finale effettuato dal proponente del software, al cui superamento segue il rilascio del prodotto ultimato.

L'importanza di un test si attua nella sua automazione, in quanto riduce il tempo dedicato alla Verifica manuale del codice, certamente più onerosa. Per questo motivo la proprietà più importante di un test è la sua ripetibilità. Per fare in modo che un test abbia questa qualità, è fondamentale definire a priori certe caratteristiche, ovvero:

- **Ambiente_G:** deve essere specificato l'insieme di componenti hardware e software su cui verrà eseguito il prodotto software, al fine di evitare problemi di incompatibilità e malfunzionamento;
- **Variabili:** si deve conoscere e garantire la corretta struttura delle variabili in ingresso ai test, in modo da prevedere gli *output_G* attesi e verificare la loro correttezza;
- **Procedure:** deve essere chiara la sequenza delle operazioni e la metodologia di applicazione dei test.

C Pianificazione dei test

Di seguito verranno visualizzate delle tabelle, strutturate secondo la sezione 5.3 delle *Norme-di-progetto-v4.2.0.pdf*, che riportano tutti i test che si sono pianificati.

C.1 Test di unità

Di seguito verrà mostrata la tabella che riporta tutti i test di unità pianificati, associati alla componente testata.

C.1.1 Descrizione dei test di unità

Sono qui riportate le tabelle per i test di unità riguardanti il client. Le funzionalità dettagliate sono descritte nella Definizione di Prodotto v.1.2.0.

Test	Descrizione	Funzioni testate	Stato
TU1	Vengono verificate tutte le funzionalità rese disponibili dal Collection Controller.	init() getData() numerify(num) previousPage() nextPage() toPage(index) changeSort() columnSort(index) delete_ document(index)	D.E.
TU2	Vengono verificate tutte le funzionalità rese disponibili dal Dashboard Controller.	CollectionListService.get	D.E.
TU3	Vengono verificate tutte le funzionalità rese disponibili dal Document Controller.	DocumentDataService.query delete_ document(index)	D.E.
TU4	Vengono verificate tutte le funzionalità rese disponibili dal DocumentEdit Controller.	DocumentEditService.query edit_ document(index) delete_ document(index)	D.E.

Tabella 2: continua nella prossima pagina

Tabella 2: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
TU5	Vengono verificate tutte le funzionalità rese disponibili dal Index Controller.	init() getData() numerify(num) previousPage() nextPage() toPage(index) changeSort() columnSort(index) delete_ document(index)	DE
TU6	Vengono verificate tutte le funzionalità rese disponibili dal Login Controller.	AuthService.login	D.E.
TU7	Vengono verificate tutte le funzionalità rese disponibili dal NavBar Controller.	CollectionListService.get logout()	D.E.
TU8	Vengono verificate tutte le funzionalità rese disponibili dal Profile Controller.	ProfileDataService.query delete_ document(index)	D.E.
TU9	Vengono verificate tutte le funzionalità rese disponibili dal ProfileEdit Controller.	ProfileEditService.query edit_ document(index) delete_ document(index)	D.E.
TU10	Vengono verificate tutte le funzionalità rese disponibili dal Query Controller.	QueryService.query createIndex(id) numerify(num) previousPage() nextPage() toPage(index) changeSort() columnSort(index) delete_ document(index)	D.E.

Tabella 2: continua nella prossima pagina

Tabella 2: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
TU11	Vengono verificate tutte le funzionalità rese disponibili dal Register Controller.	signupForm	D.E.
TU12	Vengono verificate tutte le funzionalità rese disponibili dal UserCollection Controller.	init() getData() numeryfy(num) previousPage() nextPage() toPage(index) changeSort() columnSort(index) delete_ document(index)	D.E.
TU13	Vengono verificate tutte le funzionalità rese disponibili dal UserController Controller.	UserDataService.query delete_ document(index)	D.E.
TU14	Vengono verificate tutte le funzionalità rese disponibili dal UserEdit Controller.	UserEditService.query edit_ document(index) delete_ document(index)	D.E.

Tabella 2: Test di unità parte client

Test di unità per i servizi del client.

Test	Descrizione	Stato
TU15	Vengono verificate tutte le funzionalità rese disponibili dal servizio AuthService.	D.E.
TU16	Vengono verificate tutte le funzionalità rese disponibili dal servizio CollectionDataService.	D.E.
TU17	Vengono verificate tutte le funzionalità rese disponibili dal servizio CollectionListService.	D.E.

Tabella 3: continua nella prossima pagina

Tabella 3: continua dalla pagina precedente

Test	Descrizione	Stato
TU18	Vengono verificate tutte le funzionalità rese disponibili dal servizio DocumentDataService.	D.E.
TU19	Vengono verificate tutte le funzionalità rese disponibili dal servizio DocumentEditService.	D.E.
TU20	Vengono verificate tutte le funzionalità rese disponibili dal dal servizio IndexService.	D.E.
TU21	Vengono verificate tutte le funzionalità rese disponibili dal servizio LogoutService.	D.E.
TU22	Vengono verificate tutte le funzionalità rese disponibili dal servizio ProfileDataService.	D.E.
TU23	Vengono verificate tutte le funzionalità rese disponibili dal servizio ProfileEditService.	D.E.
TU24	Vengono verificate tutte le funzionalità rese disponibili dal servizio QueryService.	D.E.
TU25	Vengono verificate tutte le funzionalità rese disponibili dal servizio RegisterService.	D.E.
TU26	Vengono verificate tutte le funzionalità rese disponibili dal servizio UserCollectionService.	D.E.
TU27	Vengono verificate tutte le funzionalità rese disponibili dal servizio UserDataService.	D.E.
TU28	Vengono verificate tutte le funzionalità rese disponibili dal servizio UserEditService.	D.E.

Tabella 3: Test di unità dei servizi del client

Sono qui riportate le tabelle per i test di unità riguardanti il server. Le funzionalità dettagliate sono descritte nella Definizione di Prodotto v.1.2.0.

Test	Descrizione	Funzioni testate	Stato
TU40	Il test verifica che il dispatcher reindirizzi in modo corretto le richieste ricevute dal client.	maap_server::controller::dispatcher::dispatcherInit()	D.E.
TU41	Il test verifica che il dispatcher venga inizializzato in maniera corretta.	maap_server::controller::frontController::initFrontController()	D.E.

Tabella 4: continua nella prossima pagina

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
TU42	Il test verifica che il front controller venga inizializzato in maniera corretta.	maap_server::controller::index:: init()	D.E.
TU43	Il test verifica che il modulo passport, utilizzato per la gestione dell'autenticazione, venga inizializzato in maniera corretta e che l'utente sia correttamente autenticato.	maap_server::controller::passport:: init-Passport() maap_server::controller::passport:: checkAuthenticatedAdmin() maap_server::controller::passport:: checkAuthenticated() maap_server::controller::passport:: checkNotAuthenticated()	superato. superato. superato. superato.
TU44	Il test verifica che la connessione al database di analisi e al database degli utenti avvenga in maniera corretta; inoltre viene verificato che venga aggiunto un utente di default all'avvio del sistema.	maap_server::modelServer::database:: index::addAdminDefault() maap_server::modelServer::database:: index::initDB()	D.E. D.E.
TU45	Il test verifica che per ogni collection definita tramite DSL esista lo schema corrispondente, e successivamente lo rende disponibile.	maap_server::modelServer::database:: MongoDBAnalysis::init()	D.E.
TU46	Il test verifica che vengano costruiti in maniera corretta gli schemi del database per l'utente e per le query; successivamente verranno creati i modelli per questi schemi e resi disponibili.	maap_server::modelServer::database:: MongoDBFramework::init()	D.E.

Tabella 4: continua nella prossima pagina

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
TU47	Il test verifica che la gestione dei dati nel database di analisi avvenga in maniera corretta. In particolare viene controllato che le collection restituite siano quelle presenti nel database, che un document possa venir modificato o rimosso. Si verifica che le query possano essere cancellate o mostrate le più utilizzate; inoltre viene verificato anche il comportamento per la creazione, rimozione o restituzione di indici.	maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: sendCollectionsList()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: sendCollection()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: sendDocument()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: sendDocumentEdit()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: updateDocument()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: removeDocument()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: resetQueries()	D.E.
		maap_server::modelServer:: data- Manager:: DatabaseAnalysisMa- nager::DatabaseAnalysisManager:: getTopQueries()	D.E.

Tabella 4: continua nella prossima pagina

[illegible]

Piano di Qualifica v4.2.0

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: applyTrasformations()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: sortDocumentsByLabels()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: getCollectionIndex()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: getDocumentShow()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: getDocumentShowEdit()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: updateDocument()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: removeDocument()	D.E.
TU49	Il test verifica la gestione di utente; in particolare viene verificata la mail dell'utente, la possibilità di effettuare la registrazione o di modificare i dati del profilo. Inoltre viene controllata la visualizzazione degli utenti quando richiesta e la possibilità di rimuovere un utente dal database.	maap_server::modelServer:: dataManager:: DatabaseUserManager::DatabaseUserManager:: checkMail()	D.E.
		maap_server::modelServer:: dataManager:: DatabaseUserManager::DatabaseUserManager:: userSignup()	D.E.

Tabella 4: continua nella prossima pagina

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
		maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::sendUserProfile() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::sendUserProfileEdit() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::updateUserProfile() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::getUsersList() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::sendUser() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::sendUserEdit() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::updateUser() maap_server::modelServer::dataManager::DatabaseUserManager::DatabaseUserManager::removeUser()	D.E. D.E. D.E. D.E. D.E. D.E. D.E.
TU50	Il test verifica che il recupero dei dati utente dal database avvenga in maniera corretta; in particolare verranno verificate le funzionalità di aggiunta utente, visualizzazione dati profilo, modifica utente e rimozione utente.	maap_server::modelServer::dataManager::DatabaseUserManager::DataRetrieverUsers::addUser() maap_server::modelServer::dataManager::DatabaseUserManager::DataRetrieverUsers::getUserProfile()	D.E. D.E.

Tabella 4: continua nella prossima pagina

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
		maap_server::modelServer:: data- Manager:: DatabaseUserMana- ger::DataRetrieverUsers::updateUserProfile() maap_server::modelServer:: data- Manager:: DatabaseUserMana- ger::DataRetrieverUsers::getUsersList() maap_server::modelServer:: data- Manager:: DatabaseUserMana- ger::DataRetrieverUsers::updateUser() maap_server::modelServer:: data- Manager:: DatabaseUserMana- ger::DataRetrieverUsers::removeUser()	D.E. D.E. D.E. D.E.
TU51	Il test verifica che le query effettuate sul database avvengano in maniera corretta. Inoltre vengono controllate le funzionalità riguardanti la creazione di indici.	maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: getModel() maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: addQuery() maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: resetQueries() maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: getQueries() maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: getIndex() maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: createIndex() maap_server::modelServer:: dataMana- ger:: IndexManager::IndexManager:: createIndex()	D.E. D.E. D.E. D.E. D.E. D.E. D.E.
TU52	Il test verifica che il JSON creato corrisponde alle richieste ricevute.	maap_server::modelServer:: dataMana- ger:: JSonComposer:: createCollection- sList() maap_server::modelServer:: dataMana- ger:: JSonComposer:: checkLabels() maap_server::modelServer:: dataMana- ger:: JSonComposer:: createCollec- tion()	superato superato superato

Tabella 4: continua nella prossima pagina

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
		maap_server::modelServer:: dataManager:: JSonComposer:: createDocument()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createQueriesList()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createIndexesList()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createUserProfile()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createUserProfileEdit()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createUsersList()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createUser()	superato
		maap_server::modelServer:: dataManager:: JSonComposer:: createUser()	superato
TU53	Il test verifica che esistono dei file DSL presenti in uno spazio apposito, e viene eseguito il parser su di essi.	maap_server::modelServer:: DSL:: DSLManager:: generateFunction()	superato
		maap_server::modelServer:: DSL:: DSLManager:: deleteFolderRecursive()	D.E.
		maap_server::modelServer:: DSL:: DSLManager:: checkDSL()	superato
TU54	Il test verifica che il parser di un file DSL venga effettuato in maniera corretta; vengono eseguiti i controlli sui campi scritti nel DSL per verificarne il contenuto.	maap_server::modelServer:: DSL:: DSLParser:: addField()	superato
		maap_server::modelServer:: DSL:: DSLParser:: checkFieldThrow()	superato
		maap_server::modelServer:: DSL:: DSLParser:: checkField()	superato
		maap_server::modelServer:: DSL:: DSLParser:: checkFieldContentThrow()	superato

Tabella 4: continua nella prossima pagina

Tabella 4: continua dalla pagina precedente

Test	Descrizione	Funzioni testate	Stato
		maap_server::modelServer:: DSL:: DSLParser:: checkFieldContent() maap_server::modelServer:: DSL:: DSLParser:: IntValue() maap_server::modelServer:: DSL:: DSLParser:: parseDSL()	superato superato superato
TU55	Il test verifica che il controllo sui file DSL parta all'avvio del server.	maap_server::modelServer:: DSL:: in- dex:: init()	superato
TU56	Il test verifica che vengono creati gli schemi giusti per le collection scritte nel DSL; inoltre viene effettuato un controllo se gli schemi sono già presenti.	maap_server::modelServer:: DSL:: schemaGenerator:: getPopulatedCollection() maap_server::modelServer:: DSL:: schemaGenerator:: arrayAddElement() maap_server::modelServer:: DSL:: schemaGenerator:: generate()	superato superato superato

Tabella 4: Test di unità parte server

C.1.2 Tracciamento

Di seguito verranno riportati in forma tabellare i tracciamenti componente-test d'unità sia lato server che lato client.

C.1.2.1 Tracciamento componente server-test di unità

Componente	Test
maap_server::controller::dispatcher:: dispatcherInit()	TU40
maap_server::controller::frontController:: initFrontController()	TU41
maap_server::controller::index:: init()	TU42
maap_server::controller::passport:: initPassport()	TU43
maap_server::controller::passport:: checkAuthenticatedAdmin()	TU43
maap_server::controller::passport:: checkAuthenticated()	TU43
maap_server::controller::passport:: checkNotAuthenticated()	TU43

Tabella 5: continua nella prossima pagina

Tabella 5: continua dalla pagina precedente

Componente	Test
maap_server::modelServer::database:: index::addAdminDefault()	TU44
maap_server::modelServer::database:: index::initDB()	TU44
maap_server::modelServer::database:: MongoDBAnalysis::init()	TU45
maap_server::modelServer::database:: MongoDBFramework::init()	TU46
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: sendCollectionsList()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: sendCollection()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: sendDocument()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: sendDocumentEdit()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: updateDocument()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: removeDocument()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: resetQueries()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: getTopQueries()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: getIndexesList()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: createIndex()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DatabaseAnalysisManager:: deleteIndex()	TU47
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: getModel()	TU48
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: getDocuments()	TU48
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: getCollectionsList()	TU48
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: applyTrasformations()	TU48
maap_server::modelServer::dataManager:: DatabaseAnalysisManager::DataRetrieverAnalysis:: sortDocumentsByLabels()	TU48

Tabella 5: continua nella prossima pagina

Tabella 5: continua dalla pagina precedente

Componente	Test
maap_server::modelServer::dataManager::DatabaseAnalysisManager::DataRetrieverAnalysis:: getCollectionIndex()	TU48
maap_server::modelServer::dataManager::DatabaseAnalysisManager::DataRetrieverAnalysis:: getDocumentShow()	TU48
maap_server::modelServer::dataManager::DatabaseAnalysisManager::DataRetrieverAnalysis:: getDocumentShowEdit()	TU48
maap_server::modelServer::dataManager::DatabaseAnalysisManager::DataRetrieverAnalysis:: updateDocument()	TU48
maap_server::modelServer::dataManager::DatabaseAnalysisManager::DataRetrieverAnalysis:: removeDocument()	TU48
maap_server::modelServer::dataManager::DatabaseUserManager:: checkMail()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: userSignup()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: sendUserProfile()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: sendUserProfileEdit()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: updateUserProfile()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: getUsersList()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: sendUser()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: sendUserEdit()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: updateUser()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager:: removeUser()	TU49
maap_server::modelServer::dataManager::DatabaseUserManager::DataRetrieverUsers:: addUser()	TU50
maap_server::modelServer::dataManager::DatabaseUserManager::DataRetrieverUsers:: getUserProfile()	TU50
maap_server::modelServer::dataManager::DatabaseUserManager::DataRetrieverUsers:: updateUserProfile()	TU50
maap_server::modelServer::dataManager::DatabaseUserManager::DataRetrieverUsers:: getUsersList()	TU50

Tabella 5: continua nella prossima pagina

Tabella 5: continua dalla pagina precedente

Componente	Test
maap_server::modelServer::dataManager:: DatabaseUserMana- ger::DataRetrieverUsers::updateUser()	TU50
maap_server::modelServer::dataManager:: DatabaseUserMana- ger::DataRetrieverUsers::removeUser()	TU50
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: getModel()	TU51
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: ad- dQuery()	TU51
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: re- setQueries()	TU51
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: ge- tQueries()	TU51
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: ge- tIndex()	TU51
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: createIndex()	TU51
maap_server::modelServer::dataManager:: IndexManager::IndexManager:: createIndex()	TU51
maap_server::modelServer::dataManager:: JJsonComposer:: createCollection- sList()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: checkLabels()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createCollection()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createDocument()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createQueriesLi- st()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createIndexesLi- st()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createUserProfi- le()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createUserProfi- leEdit()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createUsersList()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createUser()	TU52
maap_server::modelServer::dataManager:: JJsonComposer:: createUser()	TU52
maap_server::modelServer::DSL::DSLManager:: generateFunction()	TU53
maap_server::modelServer::DSL::DSLManager:: deleteFolderRecursive()	TU53

Tabella 5: continua nella prossima pagina

Tabella 5: continua dalla pagina precedente

Componente	Test
maap_server::modelServer::DSL::DSLManager:: checkDSL()	TU53
maap_server::modelServer::DSL::DSLParser:: addField()	TU54
maap_server::modelServer::DSL::DSLParser:: checkFieldThrow()	TU54
maap_server::modelServer::DSL::DSLParser:: checkField()	TU54
maap_server::modelServer::DSL::DSLParser:: checkFieldContentThrow()	TU54
maap_server::modelServer::DSL::DSLParser:: checkFieldContent()	TU54
maap_server::modelServer::DSL::DSLParser:: IntValue()	TU54
maap_server::modelServer::DSL::DSLParser:: parseDSL()	TU54
maap_server::modelServer::DSL::index:: init()	TU55
maap_server::modelServer::DSL::schemaGenerator:: getPopulatedCollection()	TU56
maap_server::modelServer::DSL::schemaGenerator:: arrayAddElement()	TU56
maap_server::modelServer::DSL::schemaGenerator:: generate()	TU56

Tabella 5: Tracciamento componente-test unità parte server

C.1.2.2 Tracciamento componente client-test di unità

Componente	Test
maap_client::controller::Collection Controller	TU1
maap_client::controller::Dashboard Controller	TU2
maap_client::controller::Document Controller	TU3
maap_client::controller::DocumentEdit Controller	TU4
maap_client::controller::Index Controller	TU5
maap_client::controller::Login Controller	TU6
maap_client::controller::NavBar Controller	TU7
maap_client::controller::Profile Controller	TU8
maap_client::controller::ProfileEdit Controller	TU9
maap_client::controller::Query Controller	TU10
maap_client::controller::Register Controller	TU11
maap_client::controller::UserCollection Controller	TU12
maap_client::controller::UserController Controller	TU13
maap_client::controller::UserEdit Controller	TU14

Tabella 6: continua nella prossima pagina

Tabella 6: continua dalla pagina precedente

Componente	Test
maap_client::services::AuthService	TU15
maap_client::services::CollectionDataService	TU16
maap_client::services::CollectionListService	TU17
maap_client::services::DocumentDataService	TU18
maap_client::services::DocumentEditService	TU19
maap_client::services::IndexService	TU20
maap_client::services::LogoutService	TU21
maap_client::services::ProfileDataService	TU22
maap_client::services::ProfileEditService	TU23
maap_client::services::QueryService	TU24
maap_client::services::RegisterService	TU25
maap_client::services::UserCollectionService	TU26
maap_client::services::UserDataService	TU27
maap_client::services::UserEditService	TU28

Tabella 6: Tracciamento componente-test unità parte client

C.2 Test di sistema

Di seguito verrà mostrata la tabella che riporta tutti i test di sistema pianificati, associati ai requisiti descritti nel documento Analisi dei Requisiti.

I test sono da intendere solo per requisiti ai quali è stato ragionevole associare un test.

C.2.1 Descrizione dei test di sistema

Test	Descrizione	Requisito	Stato
TS1	Viene verificato che il sistema MaaP generi correttamente lo scheletro necessario.	ROF1	D.E.
TS1.1	Viene verificato che il sistema MaaP installi correttamente le librerie necessarie.	ROF1.1	D.E.
TS1.2	Viene verificato che il sistema MaaP generi correttamente i file necessari.	ROF1.2	D.E.
TS1.3	Viene verificato che il sistema MaaP generi correttamente le directory necessarie.	ROF1.3	D.E.

Tabella 7: continua nella prossima pagina

Tabella 7: continua dalla pagina precedente

Test	Descrizione	Requisito	Stato
TS1.4	Viene verificato che il sottosistema di autenticazione sia installato e configurato correttamente.	ROF1.4	D.E.
TS1.4.1	Viene verificato che nel database degli utenti sia presente un profilo di amministrazione di default.	ROF1.4.1	D.E.
TS1.4.1	Viene verificato che nel database degli utenti sia presente un profilo di amministrazione di default.	ROF6	D.E.
TS1.5	Viene verificato che il sistema sia in grado di eliminare un progetto esistente.	ROF1.5	D.E.
TS1.6	Viene verificato che il sistema sia in grado di clonare un progetto esistente.	ROF1.6	D.E.
TS4	Viene verificato che il sistema crei correttamente le pagine web partendo dal loro file di descrizione.	ROF4	D.E.
TS5.1	Viene verificato che la funzione di registrazione possa essere correttamente abilitata/disabilitata.	RDF5.1	D.E.
TS5.4	Viene verificato che il sistema possa utilizzare correttamente il database di analisi.	ROF5.4	D.E.
TS5.5	Viene verificato che la funzione di creazione indici possa essere correttamente abilitata/disabilitata.	ROF5.5	D.E.
TS7	Viene verificato che il sistema consenta all'utente registrato di potersi autenticare.	ROF7	D.E.
TS8	Viene verificato che il sistema consenta all'utente di potersi registrare.	RDF8	D.E.
TS9	Viene verificato che il sistema consenta all'utente di recuperare la password.	ROF9	D.E.
TS10	Viene verificato che il sistema apra e visualizzi correttamente le Collection e le Collection-Index.	ROF10	D.E.
TS10.1	Viene verificato che il sistema visualizzi correttamente le pagine Document-Show.	ROF10.1	D.E.
TS10.2.4	Viene verificato che il sistema disconnetta correttamente un utente alla sua richiesta.	ROF10.2.4	D.E.
TS10.3.1.1	Viene verificato che un utente autenticato possa modificare i dati del suo profilo.	ROF10.3.1.1	D.E.

Tabella 7: continua nella prossima pagina

Tabella 7: continua dalla pagina precedente

Test	Descrizione	Requisito	Stato
TS10.3.1.2	Viene verificato che le modifiche apportate al profilo di un utente business autenticato siano consistenti.	ROF10.3.1.2	D.E.
TS10.3.2	Viene verificato che la creazione di un nuovo utente da parte di un utente business autenticato amministratore avvenga correttamente.	ROF10.3.2	D.E.
TS10.3.3	Viene verificata la corretta cancellazione di un utente da parte di un utente business autenticato amministratore.	ROF10.3.3	D.E.
TS10.4	Viene verificato che l'utente business autenticato amministratore possa eliminare correttamente un Document.	ROF10.4	D.E.
TS10.5	Viene verificato che l'utente business autenticato amministratore possa modificare correttamente un Document.	ROF10.5	D.E.
TS10.6	Viene verificata la corretta visualizzazione delle query più utilizzate.	ROF10.6	D.E.
TS10.7.1.2	Viene verificata la corretta creazione degli indici di analisi.	ROF10.7.1.2	D.E.
TS10.7.2	Viene verificata la corretta eliminazione degli indici di analisi.	ROF10.7.2	D.E.
TS10.7.3	Viene verificata la corretta visualizzazione degli indici di analisi.	ROF10.7.3	D.E.
TS17	Viene verificato che le pagine web prodotte dal framework MaaP siano compatibili con la versione 30.0.x o superiore di Google Chrome.	ROV18	D.E.
TS19	Viene verificato che il sistema accetti solo file di configurazione validi.	ROV20	D.E.
TS26	Viene verificato che il sistema di installazione del software funzioni correttamente.	ROV27	D.E.
TS27	Viene verificato che il deployment su Heroku avvenga con successo.	ROV28	D.E.
TS18	Viene verificato che le pagine web prodotte dal framework MaaP siano compatibili con la versione 24.x o superiore di Firefox.	ROV19	D.E.

Tabella 7: Test di sistema

C.3 Test d'integrazione

C.3.1 Diagramma d'integrazione

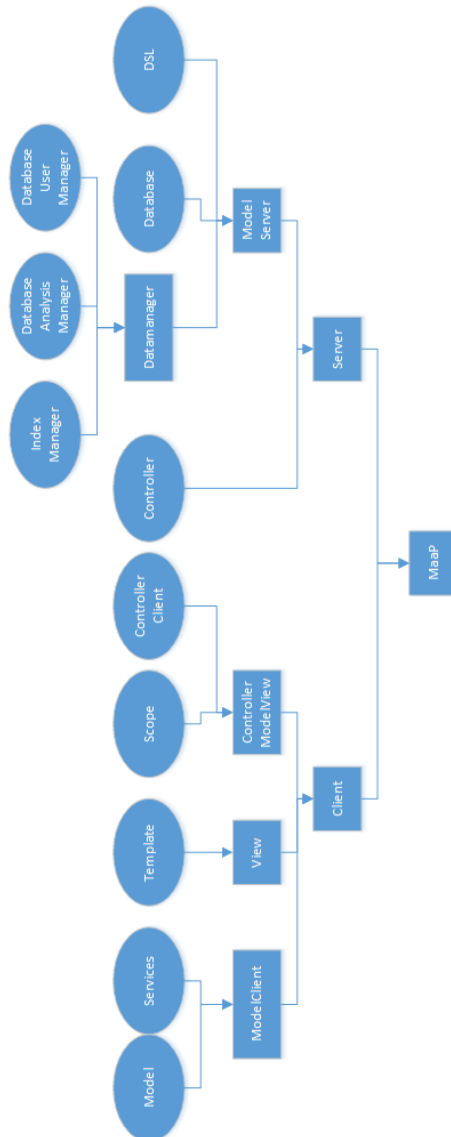


Figura 4: Diagramma d'integrazione

Di seguito verrà mostrata la tabella che riporta tutti i test d'integrazione pianificati, associati alle componenti descritte nella progettazione ad alto livello.

C.3.2 Descrizione dei test d'integrazione

Test	Descrizione	Componente	Stato
TI.MaaP	Test di integrazione finale tra client e server client e server	MaaP	D.E.
TI.Server	Verifica il corretto funzionamento delle componenti del server, quindi la corretta integrazione tra ModelServer e Controller del server. In particolare, viene verificato che le funzionalità presenti nel Controller debbano agire consistentemente sui dati del ModelServer quindi la corretta integrazione tra ModelServer e Controller del server. In particolare, viene verificato che le funzionalità presenti nel Controller debbano agire consistentemente sui dati del ModelServer.	MaaP::Server	D.E.
TI.Server.ModelServer	Verifica che tutte le operazioni di lettura, scrittura e interpretazione dei dati avvengano correttamente.	MaaP::Server::ModelServer	D.E.

Tabella 8: continua nella prossima pagina

Tabella 8: continua dalla pagina precedente

Test	Descrizione	Componente	Stato
TI.Server.ModelServer .DataManager-Database	Verifica che i gestori dei dati siano correttamente integrati ai database di riferimento siano correttamente integrati ai database di riferimento.	MaaP::Server::ModelServer ::DataManager	D.E.
TI.Server.ModelServer .DataManager-Database	Verifica che i gestori dei dati siano correttamente integrati ai database di riferimento siano correttamente integrati ai database di riferimento.	MaaP::Server::ModelServer ::Database	D.E.
TI.Server.ModelServer .DataManager	Verifica che le operazioni di recupero, gestione e scrittura dati nei database avvengano correttamente. recupero, gestione e scrittura dati nei database avvengano correttamente.	MaaP::Server::ModelServer ::DataManager	D.E.
TI.Client	Verifica che i dati ottenuti e forniti da e verso il client siano corretti e correttamente gestiti all'interno del client stesso corretti e correttamente gestiti all'interno del client stesso.	MaaP::Client	D.E.
TI.Client.View-ControllerModelView	Verifica che i dati uscenti ed entranti la View siano gestiti correttamente dal ControllerModelView	MaaP::Client::View	D.E.

Tabella 8: continua nella prossima pagina

Tabella 8: continua dalla pagina precedente

Test	Descrizione	Componente	Stato
TI.Client.View-ControllerModelView	Verifica che i dati uscenti ed entranti la View siano gestiti correttamente dal ControllerModelView	MaaP::Client::ControllerModelView	D.E.
TI.Client.ControllerModelView-ModelClient	Viene verificato che le funzionalità del ControllerModelView agiscano correttamente sul ModelClient	MaaP::Client::ControllerModelView	D.E.
TI.Client.ControllerModelView	Viene verificato che le funzioni del ClientController modifichino correttamente i dati presenti nello Scope nello Scope	MaaP::Client::ModelClient	D.E.
TI.Client.ControllerModelView	Viene verificato che le funzioni del ClientController modifichino correttamente i dati presenti nello Scope nello Scope	MaaP::Client::ControllerModelView	D.E.

Tabella 8: Test d'integrazione

C.3.3 Tracciamento

Di seguito verranno riportati in forma tabellare i tracciamenti componente-test d'integrazione e test d'integrazione-componente.

C.3.3.1 Tracciamento componente-test d'integrazione

Componente	Test
MaaP::Client	TI.Client
MaaP::Client::ControllerModelView	TI.Client.ControllerModelView

MaaP::Client::ModelClient	TI.Client.ControllerModelView
MaaP::Client::ControllerModelView	TI.Client.ControllerModelView-ModelClient TI.Client.View-ControllerModelView
MaaP::Client::View	TI.Client.View-ControllerModelView
MaaP	TI.MaaP
MaaP::Server	TI.Server
MaaP::Server::ModelServer	TI.Server.ModelServer TI.Server.ModelServer.DataManager TI.Server.ModelServer.DataManager- Database TI.Server.ModelServer.DataManager- Database

Tabella 9: Tracciamento componente-test d'integrazione

C.3.3.2 Tracciamento test d'integrazione-componente

Test	Componente
TI.MaaP	MaaP
TI.Server	MaaP::Server
TI.Server.ModelServer	MaaP::Server::ModelServer MaaP::Server::ModelServer::DataManager MaaP::Server::ModelServer::Database MaaP::Server::ModelServer::DataManager
TI.Client	MaaP::Client
TI.Client.View-ControllerModelView	MaaP::Client::View
TI.Client.ControllerModelView-ModelClient	MaaP::Client::ControllerModelView MaaP::Client::ModelClient
TI.Client.View-ControllerModelView	MaaP::Client::ControllerModelView

| TI.Client.ControllerModelView

| MaaP::Client::ControllerModelView

Tabella 10: Tracciamento test d'integrazione-componente

C.4 Test di validazione

In questa sezione vengono descritti i test di validazione, utili ad accertarsi che il prodotto realizzato sia conforme alle attese. Per ogni test viene descritta la serie di passi che un utente deve eseguire per validare la conformità ai requisiti ad esso associati.

C.4.1 TV 1 - Lato Utente Business

L'utente vuole verificare le funzionalità di visualizzazione di Collection e Document, per tanto gli è richiesto di:

1. Cliccare sul pulsante Collection;
2. Selezionare la Collection da visualizzare dall'apposito elenco;
 - (a) Scorrere le pagine della Collection fino a trovare il Document ricercato;
 - (b) Usare un *filtro_G* per ricercare un Document;
 - (c) Annullare il filtro applicato;
 - (d) Ordinare una *chiave_G*, se abilitata, in ordine crescente o decrescente;
 - (e) Impostare un numero massimo di Document da visualizzare.
3. Selezionare il Document ricercato cliccando sulla chiave selezionabile;
4. Verificare che il Document venga visualizzato correttamente.

C.4.2 TV 2 - Lato Utente Business

L'utente vuole verificare il corretto funzionamento delle funzionalità di registrazione, autenticazione e gestione profilo. Si dovrà quindi seguire la seguente *procedura_G* di validazione:

1. Verificare di potersi registrare;
 - (a) Cliccare sul pulsante di registrazione;
 - (b) Compilare il *form_G* di registrazione con dati validi;
 - (c) Confermare la registrazione mediante il *link_G* contenuto nella mail inviata automaticamente dal sistema.
2. Verificare di poter recuperare la password in caso di dimenticanza;
 - (a) Cliccare sul pulsante di recupero password;
 - (b) Inserire la *email_G* con cui si è registrati al sistema;
 - (c) Verificare che il sistema invii correttamente la mail per il recupero della password;
 - (d) Verificare che la nuova password fornita nella mail consenta l'accesso al sistema.
3. Verificare di poter effettuare l'autenticazione;

- (a) Cliccare sul pulsante di autenticazione;
 - (b) Inserire le credenziali precedentemente registrate;
 - (c) Confermare l'autenticazione.
4. Verificare che il sistema effettui correttamente l'autenticazione;
5. Verificare di poter modificare i propri dati utente;
- (a) Entrare nel proprio profilo utente;
 - (b) Cliccare sul pulsante di modifica;
 - (c) Verificare di poter modificare una o più chiavi del profilo;
 - (d) Verificare di poter salvare le modifiche;
 - (e) Verificare di poter annullare le modifiche.
6. Verificare di potersi disconnettere correttamente dal sistema;
- (a) Cliccare sul pulsante di disconnessione;
 - (b) Verificare di non poter più accedere a pagine riservate ad utenti autenticati.

C.4.3 TV 3 - Lato Utente Business Amministratore

L'utente amministratore vuole verificare il corretto funzionamento della gestione dei profili utenti. Si dovrà quindi seguire la seguente procedura:

1. Effettuare l'autenticazione con le proprie credenziali;
 - (a) Al primo accesso assoluto al sistema, utilizzare le credenziali di default fornite dal sistema stesso.
2. Cliccare sul pulsante di gestione profili utenti;
3. Selezionare il profilo utente da modificare;
4. Verificare di poter modificare le informazioni del profilo utente;
 - (a) Cliccare sul pulsante di modifica;
 - (b) Verificare di poter modificare una o più chiavi del profilo;
 - (c) Verificare di poter salvare le modifiche;
 - (d) Verificare di poter annullare le modifiche.
5. Verificare di poter modificare i permessi associati al profilo utente;
 - (a) Cliccare sul pulsante di modifica;
 - (b) Verificare di poter modificare i permessi disponibili;
 - (c) Verificare di poter salvare le modifiche;
 - (d) Verificare di poter annullare le modifiche.
6. Verificare di poter eliminare correttamente un utente;
 - (a) Entrare nel profilo utente da eliminare;
 - (b) Cliccare sul pulsante di eliminazione;

- (c) Confermare l'eliminazione;
 - (d) Verificare che le credenziali dell'utente cancellato non siano più presenti nel sistema.
7. Verificare di poter creare un nuovo profilo utente;
- (a) Cliccare sul pulsante di creazione;
 - (b) Completare il form di registrazione con informazioni valide;
 - (c) Verificare che il profilo sia stato creato correttamente.

C.4.4 TV 4 - Lato Utente Business Amministratore

L'utente amministratore vuole verificare il corretto funzionamento della gestione dei Document. Dovrà quindi seguire la seguente procedura:

1. Dopo essere entrato in una pagina Collection, selezionare un Document;
2. Verificare di poter modificare un Document;
 - (a) Cliccare sul pulsante di modifica;
 - (b) Verificare di poter modificare le chiavi disponibili;
 - (c) Verificare di poter salvare le modifiche;
 - (d) Verificare di poter annullare le modifiche.
3. Verificare di poter cancellare un Document;
 - (a) Selezionare il pulsante di cancellazione;
 - (b) Verificare che il Document non sia più presente nel sistema.

C.4.5 TV 5 - Lato Utente Business Amministratore

L'utente amministratore vuole verificare il corretto funzionamento della gestione degli indici. Dovrà quindi seguire la seguente procedura:

1. Entrare nella gestione degli indici;
2. Verificare di poter creare un indice;
 - (a) Cliccare sul pulsante di creazione indici;
 - (b) Selezionare una tra le query più utilizzate;
 - (c) Confermare la creazione;
 - (d) Verificar che l'indice sia stato creato correttamente.
3. Verificare di poter eliminare un indice;
 - (a) Selezionare un indice presente;
 - (b) Cliccare sul pulsante di eliminazione;
 - (c) Verificare che l'indice non sia più presente.

C.4.6 TV 6 - Lato Utente Sviluppatore

L'utente sviluppatore vuole verificare il corretto funzionamento della gestione ed interpretazione del *DSL_G* per la creazione di una pagina Collection-Index. Dovrà quindi seguire la seguente procedura:

1. Scrivere un file DSL valido;
 - (a) Specificare le chiavi da visualizzare;
 - (b) Specificare la provenienza dei dati;
 - i. Provenienti da query standard;
 - ii. Provenienti da query personalizzate;
 - iii. Provenienti da funzioni personalizzate;
 - iv. Provenienti da riferimenti esterni.
 - (c) Specificare eventuali etichette;
 - (d) Specificare eventuali chiavi ordinabili;
 - (e) Specificare il numero massimo di Document per pagina;
 - (f) Specificare il nome della Collection;
 - (g) Specificare la posizione della Collection nel menù.
2. Caricare il file DSL nell'apposita cartella;
3. Avviare il server MaaP;
4. Effettuare l'autenticazione;
5. Entrare nella Collection la cui pagina è descritta dal file DSL inserito e verificare la bontà della sua interpretazione.

C.4.7 TV 7 - Lato Utente Sviluppatore

L'utente sviluppatore vuole verificare il corretto funzionamento della gestione ed interpretazione del DSL per la creazione di una pagina Document-Show. Dovrà quindi seguire la seguente procedura:

1. Scrivere un file DSL valido;
 - (a) Specificare le chiavi da visualizzare;
 - (b) Specificare la provenienza dei dati;
 - i. Provenienti da query standard;
 - ii. Provenienti da query personalizzate;
 - iii. Provenienti da funzioni personalizzate;
 - iv. Provenienti da riferimenti esterni.
 - (c) Specificare eventuali etichette per le chiavi;
2. Caricare il file DSL nell'apposita cartella;
3. Avviare il server MaaP;
4. Effettuare l'autenticazione;
5. Entrare nel document la cui pagina è descritta dal file DSL inserito e verificare la bontà della sua interpretazione.

C.4.8 TV 8 - Lato Utente Sviluppatore

L'utente sviluppatore vuole verificare il corretto funzionamento delle funzioni di gestione del progetto. Dovrà quindi seguire la seguente procedura:

1. Creare un nuovo progetto;
2. Verificare che il nuovo progetto sia disponibile;
3. Clonare un progetto;
4. Verificare che il progetto clonato sia identico all'originale;
5. Verificare di poter gestire un progetto;
 - (a) Verificare di poter avviare, fermare o riavviare il server MaaP;
 - (b) Verificare di poter modificare i file di configurazione del progetto;
 - i. Abilitare/disabilitare la registrazione;
 - ii. Abilitare/disabilitare la creazione di indici;
 - iii. Impostare la connessione al database di analisi.
 - (c) Verificare di poter modificare o caricare nuovi file DSL;
 - (d) Verificare di poter modificare i *template_G* disponibili.
6. Eliminare un progetto;
7. Verificare che il progetto eliminato non sia più disponibile.

C.4.9 Tracciamento

Di seguito verranno riportati in forma tabellare i tracciamenti test di validazione - requisiti e l'inverso, ovvero requisiti - test di validazione.

C.4.9.1 Tracciamento Test di Validazione - Requisiti

Test	Requisiti
TV1	RDF10.2 RDF10.2.1 RDF10.2.1.1 RDF10.2.1.2 RDF10.2.2 RDF10.2.3 ROF10 ROF10.1 ROF10.2.5
TV2	RDF8 RDF8.1 RDF8.2

Tabella 11: continua nella prossima pagina

Tabella 11: continua dalla pagina precedente

Test	Requisiti
	RDF8.2.1 ROF10.2.4 ROF10.3 ROF10.3.1 ROF10.3.1.1 ROF10.3.1.2 ROF10.3.1.3 ROF7 ROF7.1 ROF7.2 ROF7.2.1 ROF9
TV3	ROF10.3.1.4 ROF10.3.2 ROF10.3.3 ROF6
TV4	ROF10.4 ROF10.5 ROF10.5.1 ROF10.5.2 ROF10.5.3
TV5	ROF10.6 ROF10.7 ROF10.7.1 ROF10.7.1.1 ROF10.7.1.2 ROF10.7.2 ROF10.7.2.1 ROF10.7.2.2 ROF10.7.3
TV6	RFF4.2.2.7 ROF3 ROF4 ROF4.1 ROF4.1.2 ROF4.1.2.1 ROF4.1.2.1.1 ROF4.1.2.1.2 ROF4.1.2.1.3 ROF4.1.2.1.4 ROF4.1.2.1.5 ROF4.1.2.2 ROF4.1.2.3 ROF4.2

Tabella 11: continua nella prossima pagina

Tabella 11: continua dalla pagina precedente

Test	Requisiti
	ROF4.2.1 ROF4.2.1.1 ROF4.2.1.2 ROF4.2.2 ROF4.2.2.1 ROF4.2.2.2 ROF4.2.2.3 ROF4.2.2.4 ROF4.2.2.5 ROF4.2.2.6 ROF4.3 ROF4.4
TV7	ROF4.1.3 ROF4.1.3.1 ROF4.1.3.1.1 ROF4.1.3.1.2 ROF4.1.3.1.3 ROF4.1.3.1.4 ROF4.1.3.1.5 ROF4.2.3 ROF4.2.3.1 ROF4.2.3.2 ROF4.3 ROF4.4
TV8	RDF5.1 RDF5.3 ROF5 ROF5.4 ROF5.5

Tabella 11: Tracciamento Test di validazione - Requisiti

C.4.9.2 Tracciamento Requisiti - Test di Validazione

Requisiti	Test
RDF10.2.3	TV1
ROF10	TV1
ROF10.2.5	TV1
RDF10.2.2	TV1

Tabella 12: continua nella prossima pagina

Tabella 12: continua dalla pagina precedente

Requisiti	Test
RDF10.2.1.2	TV1
RDF10.2.1.1	TV1
RDF10.2.1	TV1
RDF10.2	TV1
ROF10.1	TV1
ROF10.3.1	TV2
RDF8.2.1	TV2
ROF10.3.1.1	TV2
ROF10.3	TV2
ROF10.2.4	TV2
ROF9	TV2
ROF10.3.1.3	TV2
RDF8.1	TV2
RDF8	TV2
ROF7.2.1	TV2
ROF7.2	TV2
ROF7.1	TV2
ROF7	TV2
RDF8.2	TV2
ROF10.3.1.2	TV2
ROF6	TV3
ROF10.3.1.4	TV3
ROF10.3.2	TV3
ROF10.3.3	TV3
ROF10.4	TV4
ROF10.5.3	TV4
ROF10.5.2	TV4
ROF10.5	TV4
ROF10.5.1	TV4
ROF10.7.3	TV5
ROF10.6	TV5
ROF10.7	TV5

Tabella 12: continua nella prossima pagina

Tabella 12: continua dalla pagina precedente

Requisiti	Test
ROF10.7.1	TV5
ROF10.7.1.1	TV5
ROF10.7.1.2	TV5
ROF10.7.2	TV5
ROF10.7.2.1	TV5
ROF10.7.2.2	TV5
ROF4.2.2.4	TV6
ROF4.4	TV6
ROF4.3	TV6
RFF4.2.2.7	TV6
ROF3	TV6
ROF4.2.2.5	TV6
ROF4	TV6
ROF4.2.2.3	TV6
ROF4.2.2.2	TV6
ROF4.2.2.1	TV6
ROF4.2.2	TV6
ROF4.2.1.2	TV6
ROF4.2.1.1	TV6
ROF4.1.2.1.1	TV6
ROF4.2.2.6	TV6
ROF4.2.1	TV6
ROF4.1	TV6
ROF4.1.2.1	TV6
ROF4.1.2.1.2	TV6
ROF4.1.2.1.3	TV6
ROF4.1.2.1.4	TV6
ROF4.1.2.1.5	TV6
ROF4.1.2.2	TV6
ROF4.1.2.3	TV6
ROF4.2	TV6
ROF4.1.2	TV6

Tabella 12: continua nella prossima pagina

Tabella 12: continua dalla pagina precedente

Requisiti	Test
ROF4.1.3.1	TV7
ROF4.2.3	TV7
ROF4.4	TV7
ROF4.3	TV7
ROF4.2.3.2	TV7
ROF4.2.3.1	TV7
ROF4.1.3.1.5	TV7
ROF4.1.3.1.4	TV7
ROF4.1.3.1.3	TV7
ROF4.1.3.1.1	TV7
ROF4.1.3	TV7
ROF4.1.3.1.2	TV7
ROF5.5	TV8
ROF5	TV8
RDF5.1	TV8
RDF5.3	TV8
ROF5.4	TV8

Tabella 12: Tracciamento Requisiti - Test di validazione

D Gestione amministrativa della revisione

Di seguito verrà descritto come avviene, all'interno del gruppo, la comunicazione per la gestione di anomalie e per il trattamento delle discrepanze.

D.1 Comunicazione e risoluzione di anomalie

Con anomalia si intende un esito diverso del prodotto rispetto alle aspettative, una violazione delle norme tipografiche di un documento, un valore di qualche indice non valido, ovvero fuori dal range di accettazione. Se un verificatore scova un'anomalia, di conseguenza aprirà un *ticket_G* su RedMine, strumento descritto nelle *Norme di progetto v4.2.0.pdf*, in appendice A.1.

D.2 Trattamento delle discrepanze

La discrepanza indica una mancata corrispondenza tra il prodotto atteso e il prodotto finito. Essa non ostruisce il funzionamento del software, ma è inesatto rispetto ai requisiti descritti. Per la gestione delle discrepanze si procede nella stessa maniera vista per la gestione delle anomalie.

E Resoconto delle attività di verifica

E.1 Resoconto PDCA

Durante lo sviluppo del progetto si è sempre applicato il ciclo di Deming per cercare di migliorare la qualità dei processi. Questa attività ha portato a molteplici miglioramenti. Tra i più significativi si riportano:

- Migliore politica di assegnazione delle attività:

All'inizio della progettazione, le attività principali e maggiormente propedeutiche erano state assegnate a gruppi di quattro persone, con l'obiettivo di colmare velocemente il monte ore/-persona e ridurre i tempi di completamento. Tuttavia si è notato come il lavoro procedesse più lentamente di quanto atteso e fossero presenti dei ritardi rispetto alle consegne. Nelle seguenti pianificazioni si è ridotto il numero di persone assegnate ad una singola attività cercando nel contempo di massimizzare il parallelismo. Nonostante questo abbia leggermente aumentato il tempo di completamento delle singole attività, i benefici portati dal lavoro concorrente di team più piccoli hanno largamente sorpassato gli svantaggi e comportato un maggior avanzamento complessivo del progetto.

- Migliore sviluppo dei documenti:

All'inizio i documenti erano sviluppati utilizzando *editor_G* di testo come OpenOffice e in seguito, una volta completati, trasformati in LATEX con una formattazione appropriata. In seguito si è visto come lo sviluppo dei documenti direttamente in LATEX , nonostante sia più oneroso in termini di tempo durante la stesura, sia comunque più rapido rispetto al primo metodo e, soprattutto, privo degli errori di trasposizione, riducendo quindi le ore/persona necessarie e gli errori contenuti nei documenti.

E.2 Riassunto delle attività di verifica

In questa sezione sono descritti i resoconti delle attività di verifica effettuate sui documenti prima di ciascuna revisione.

E.2.1 Revisione dei Requisiti

Nel periodo precedente a questa revisione i documenti sono stati controllati dai Verificatori seguendo le *Norme_di_progetto_v4.2.0.pdf*; è stata applicata l'analisi statica descritta nella sezione 2.8.1 di questo documento. Inizialmente è stata applicata la tecnica di Walkthrough, dove sono scovati e successivamente corretti gli errori; ogni volta che si trovava un errore, esso veniva messo nell'apposta lista che serve per l'Inspection. Dopo il Walkthrough è stata applicata la tecnica di Inspection, utilizzando l'apposita lista, disponibile in appendice delle Norme di Progetto. Inoltre per questo documento sono state calcolate le metriche descritte nella sezione 2.9.2 del documento corrente. Per quanto riguarda i processi, essi sono stati controllati e verificati secondo le metodologie descritte nelle *Norme_di_progetto_v4.2.0.pdf*. Sono state calcolate le metriche per i processi descritti in sezione 2.9.1 di questo documento, e riportati i corrispondenti valori di BV e SV in forma tabellare.

E.2.2 Revisione di Progettazione

Nel periodo precedente a questa revisione i documenti sono stati controllati dai verificatori seguendo le *Norme_di_progetto_v4.2.0.pdf*; è stata applicata l'analisi statica descritta nella sezione 2.8.1 di questo documento. Inizialmente è stata applicata la tecnica di Walkthrough, dove sono scovati e successivamente corretti gli errori; ogni volta che si trovava un errore, esso veniva messo nell'apposta

lista che serve per l'Inspection. Dopo il Walkthrough è stata applicata la tecnica di Inspection, utilizzando l'apposita lista, disponibile in appendice delle Norme di Progetto; è stata posta particolare attenzione al documento Specifica Tecnica. Inoltre per questo documento sono state calcolate le metriche descritte nella sezione 2.9.2 del documento corrente. Per quanto riguarda i processi, essi sono stati controllati e verificati secondo le metodologie descritte nelle *Norme_di_progetto_v4.2.0.pdf*. Sono state calcolate le metriche per i processi descritti in sezione 2.9.1 di questo documento, e riportati i corrispondenti valori di BV e SV in forma tabellare.

E.2.3 Revisione di Qualifica

Nel periodo precedente a questa revisione i documenti sono stati controllati dai verificatori seguendo le *Norme_di_progetto_v4.2.0.pdf*; è stata applicata l'analisi statica descritta nella sezione 2.8.1 di questo documento. Inizialmente è stata applicata la tecnica di Walkthrough, dove sono scovati e successivamente corretti gli errori; ogni volta che si trovava un errore, esso veniva messo nell'apposta lista che serve per l'Inspection. Dopo il Walkthrough è stata applicata la tecnica di Inspection, utilizzando l'apposita lista, disponibile in appendice delle Norme di Progetto. E' stata posta particolare attenzione al documento Specifica Tecnica. Inoltre per questo documento sono state calcolate le metriche descritte nella sezione 2.9.2 del documento corrente. Per quanto riguarda i processi, essi sono stati controllati e verificati secondo le metodologie descritte nelle *Norme_di_progetto_v4.2.0.pdf*. Sono state calcolate le metriche per i processi descritti in sezione 2.9.1 di questo documento, e riportati i corrispondenti valori di BV e SV in forma tabellare.

E.3 Dettaglio delle verifiche tramite analisi

E.3.1 Analisi dei Requisiti

E.3.1.1 Processi Di seguito vengono riportati i valori degli indici SV e BV calcolati durante il periodo di tempo dedicato all'Analisi dei Requisiti.

Attività	SV	BV
<i>Studio Fattibilità</i>	0	0
<i>Analisi dei Requisiti</i>	+50	+50
<i>Glossario</i>	0	0
<i>Norme di Progetto</i>	0	0
<i>Piano di Progetto</i>	0	0
<i>Piano di Qualifica</i>	-15	-15

Tabella 13: BV e SV calcolati sui documenti durante l'Analisi

E.3.1.2 Conclusioni In questa tabella, i valori positivi indicano un costo risparmiato, viceversa i valori negativi mostrano un costo eccedente. I valori indicati in tabella sono espressi in euro. Non avendo previsto degli intervalli di tempo libero tra un'attività e la successiva, abbiamo ottenuto degli SV positivi in Analisi dei Requisiti e negativi in Piano di Qualifica. Questa è stata una mancanza da parte del team, che vedrà di migliorarsi nelle prossime fasi e di adottare una tattica di pianificazione più flessibile. I costi aggiuntivi sono comunque in linea con i nostri obiettivi.

E.3.1.3 Documenti Di seguito vengono riportati, per ogni documento, i valori dell'indice di Gulpease calcolati durante il periodo di tempo dedicato all'Analisi dei Requisiti. Un documento è valido solo se rispecchia i range in sezione 2.9.2.1.

Documento	Valore indice	Esito
<i>Studio Fattibilità v1.2.0</i>	46	Sufficiente
<i>Analisi dei Requisiti v1.2.0</i>	52	Superato
<i>Glossario v1.2.0</i>	46	Sufficiente
<i>Norme di Progetto v1.2.0</i>	52	Superato
<i>Piano di Progetto v1.2.0</i>	50	Superato
<i>Piano di Qualifica v1.2.0</i>	47	Sufficiente

Tabella 14: Esiti dell'indice di Gulpease calcolato sui documenti durante l'Analisi

E.3.2 Analisi in Dettaglio

E.3.2.1 Processi Di seguito vengono riportati i valori degli indici SV e BV calcolati durante il periodo di tempo dedicato all'Analisi in Dettaglio.

Attività	SV	BV
<i>Studio Fattibilità</i>	0	0
<i>Analisi dei Requisiti</i>	0	-50
<i>Glossario</i>	0	0
<i>Norme di Progetto</i>	0	0
<i>Piano di Progetto</i>	0	0
<i>Piano di Qualifica</i>	0	0

Tabella 15: BV e SV calcolati sui documenti durante l'Analisi in Dettaglio

E.3.2.2 Conclusioni Come si può notare dalla tabella, il BV è negativo, in quanto non sono state pianificate alcune attività correttive, ed è stato messo a budget il costo necessario per effettuare queste attività non previste.

Lo SV invece è pari a zero, in quanto l'ampio slack di tempo pianificato è servito a coprire le correzioni non previste e di conseguenza non è stato prodotto niente di più rispetto a quanto pianificato.

E.3.2.3 Documenti Di seguito vengono riportati, per ogni documento, i valori dell'indice di Gulpease calcolati durante il periodo di tempo dedicato all'Analisi in Dettaglio.

Documento	Valore indice	Esito
<i>Analisi dei Requisiti v2.2.0</i>	55	Superato
<i>Glossario v2.2.0</i>	56	Sufficiente
<i>Norme di Progetto v2.2.0</i>	52	Superato

<i>Piano di Progetto v2.2.0</i>	48	Sufficiente
<i>Piano di Qualifica v2.2.0</i>	47	Sufficiente

Tabella 16: Esiti dell'indice di Gulpease calcolato sui documenti durante l'Analisi in Dettaglio

E.3.3 Progettazione Architettuale

E.3.3.1 Processi Di seguito vengono riportati i valori degli indici SV e BV calcolati durante il periodo di tempo dedicato alla Progettazione Architettuale.

Attività	SV	BV
<i>Analisi dei Requisiti</i>	0	+35
<i>Glossario</i>	0	+45
<i>Norme di Progetto</i>	20	+45
<i>Piano di Progetto</i>	0	+45
<i>Piano di Qualifica</i>	0	-20
<i>Specifica Tecnica</i>	0	-87

Tabella 17: BV e SV calcolati sui documenti durante la Progettazione Architettuale

E.3.3.2 Conclusioni Lo SV è positivo, in quanto lo slack dedicato al documento Norme di Progetto ha *permesso_G* l'aggiunta di valore non pianificato, come l'aggiunta di sezioni.

Il BV è positivo, e nonostante il fatto che si è dedicato più tempo alla progettazione, e quindi dedicandoci più budget; a causa di questo si è riuscito a risparmiare budget per le attività dedicate agli altri documenti, dedicando maggior budget per la Verifica della progettazione, che nella pianificazione non era adeguato, e togliendone da altre attività.

E.3.3.3 Documenti Di seguito vengono riportati, per ogni documento, i valori dell'indice di Gulpease calcolati durante il periodo di tempo dedicato alla Progettazione Architettuale.

Documento	Valore indice	Esito
<i>Analisi dei Requisiti v3.2.0</i>	58	Superato
<i>Glossario v3.2.0</i>	56	Superato
<i>Norme di Progetto v3.2.0</i>	53	Superato
<i>Piano di Progetto v3.2.0</i>	49	Sufficiente
<i>Piano di Qualifica v3.2.0</i>	48	Sufficiente
<i>Specifica Tecnica v1.2.0</i>	44	Sufficiente

Tabella 18: Esiti dell'indice di Gulpease calcolato sui documenti durante la Progettazione

E.3.3.4 Progettazione Viene qui riportata una tabella riassuntiva che riporta il calcolo dei parametri di accoppiamento afferente ed efferente per i componenti individuati nella progettazione architettuale.

Componente	Afferente	Efferente
MaaP::Server	1	0
MaaP::Server::ModelServer	3	0
MaaP::Server::ModelServer::DataManager	1	4
MaaP::Server::ModelServer::DataManager::DatabaseAnalysisManager	1	7
MaaP::Server::ModelServer::DataManager::DatabaseUserManager	1	4
MaaP::Server::ModelServer::DataManager::IndexManager	1	2
MaaP::Server::ModelServer::Database	4	0
MaaP::Server::ModelServer::DSL	1	0
MaaP::Server::Controller	1	4
MaaP::Client	0	1
MaaP::Client::View	0	14
MaaP::Client::ControllerModelView	14	22
MaaP::Client::ControllerModelView::ControllerClient	14	22
MaaP::Client::ModelClient	22	1
MaaP::Client::ModelClient::Services	20	1
MaaP::Client::ModelClient::Directives	2	0
MaaP::Client::ModelClient::Model	0	0

Tabella 19: Tabella accoppiamento componenti

Come si può vedere dalla tabella, l'accoppiamento afferente risulta generalmente basso ad eccezione del componente ControllerModelView del package Client e relativo ControllerClient i quali hanno un valore relativamente alto. Questo delinea la criticità del componente in oggetto, che quindi andrà trattato con dovute cautele durante la generazione dei test e la loro esecuzione per ottenere un componente stabile più velocemente, prevenendo il rischio di regressione dovuto ad un alto accoppiamento.

Per quanto riguarda l'accoppiamento efferente, anch'esso è relativamente basso ad eccezione dei componenti interni del package MaaP::Server::ModelServer::DataManager e del componente ControllerClient che per la loro natura intrinseca hanno un alto livello di accoppiamento dovendo interagire con diverse classi di package esterni.

E.3.4 Progettazione di dettaglio e codifica

E.3.4.1 Processi Di seguito vengono riportati i valori degli indici SV e BV calcolati durante il periodo di tempo dedicato alla Progettazione di dettaglio e codifica.

Attività	SV	BV
<i>Analisi dei Requisiti</i>	0	+25
<i>Glossario</i>	0	-15

<i>Norme di Progetto</i>	0	-30
<i>Piano di Progetto</i>	0	+90
<i>Piano di Qualifica</i>	0	+51
<i>Specifica Tecnica</i>	0	-1
<i>Definizione di Prodotto</i>	0	-241
<i>Manuale utente</i>	-105	+105
<i>Codifica</i>	0	+480
<i>Verifica della Codifica</i>	0	+150

Tabella 20: BV e SV calcolati sui documenti durante la Progettazione di dettaglio e codifica

E.3.4.2 Conclusioni Lo SV è negativo in quanto non è stato completato il Manuale Utente come pianificato. Il BV è invece positivo a causa del fatto che sono state richieste molte meno ore di Codifica per realizzare quanto pianificato, e di conseguenza sono state tolte ore di Verifica del codice per il motivo precedente. Inoltre il fatto di non aver completato il Manuale Utente alza il BV, a discapito dello SV; il BV della Definizione di Prodotto è negativo perchè sono state aggiunte ore di Verifica non pianificate; nonostante questo, il BV complessivo è positivo.

E.3.4.3 Documenti Di seguito vengono riportati, per ogni documento, i valori dell'indice di Gulpease calcolati durante il periodo di tempo dedicato alla Progettazione di dettaglio e codifica.

Documento	Valore indice	Esito
<i>Analisi dei Requisiti v4.2.0</i>	58	Superato
<i>Glossario v4.2.0</i>	57	Superato
<i>Norme di Progetto v4.2.0</i>	56	Superato
<i>Piano di Progetto v4.2.0</i>	51	Superato
<i>Piano di Qualifica v4.2.0</i>	45	Sufficiente
<i>Specifica Tecnica v2.2.0</i>	52	Superato
<i>Definizione di Prodotto v1.2.0</i>	62	Superato
<i>Manuale Utente v1.2.0</i>	83	Superato

Tabella 21: Esiti dell'indice di Gulpease calcolato sui documenti durante la Progettazione di dettaglio e codifica

E.4 Dettaglio dell'esito delle revisioni

Per ciascuna revisione alla quale si intende partecipare, il Committente avrà il compito di segnalare eventuali problematiche trovate, dando una valutazione globale dell'andamento del progetto e una descrizione per ciascun documento con correzioni e accorgimenti da apportare. Di seguito vengono elencate le modifiche apportate ai documenti, come suggerito dal Committente, per ciascuna revisione.

E.4.1 Revisione dei Requisiti

- **Studio di Fattibilità:** il documento ha avuto una valutazione positiva, quindi non ci sono stati accorgimenti da apportare;
- **Norme di Progetto:** il documento è stato riorganizzato come suggerito, ovvero per processi, attività procedure e strumenti; è stata migliorata la descrizione della rotazione dei ruoli e il documento è stato incrementato con le parti riguardanti la parte di progettazione;
- **Analisi dei Requisiti:** sono stati corretti degli errori grammaticali, chiariti i significati di alcune parole; i casi d'uso segnalati hanno subito modifiche e aggiustamenti alle pre e post condizioni, mentre altri sono stati descritti più approfonditamente. Sempre dei casi d'uso sono stati tolti o spostati perché in contrasto tra di loro, mentre per quanto riguarda la suddivisione dei requisiti in funzionali, desiderabili, obbligatori ecc.. sono stati rimossi e spostati perché non adatti alla categoria in cui si presentavano. Il documento ha avuto una buona valutazione sulla struttura, quindi non si è cambiata.
- **Piano di Progetto:** come suggerito, alcuni contenuti sono stati spostati nell'Appendice del documento; è stato corretto l'utilizzo della parola fase, e usata solo se strettamente necessario e in contesti che la richiedono. La sezione Preventivo a finire è stata corretta in Consuntivo, in quanto si è capito la differenza tra i significati dei due termini. Si è deciso, per i prossimi intervalli di tempo antecedenti le revisioni, di dedicare più tempo all'attività di Verifica, cercando di raggiungere la soglia del 30% del tempo totale, come suggerito. Il documento inoltre è stato incrementato con le parti relative alla progettazione;
- **Piano di Qualifica:** il documento ha subito profonde modifiche, è stato ristrutturato e riorganizzato. Per fare ciò, è stata seguita la best practice per la struttura dei documenti presente nel sito del Professor Vardanega; il documento ha subito profonde modifiche anche nei contenuti, inoltre è stato incrementato con le parti relative alla progettazione;
- **Glossario:** il documento ha subito una lieve ristrutturazione, è stato tolto l'indice come suggerito; il documento è stato incrementato con l'inserimento di altri termini.

E.4.2 Revisione di Progettazione

- **Norme di Progetto:** il documento è stato riorganizzato come suggerito ed è stato incrementato con le parti riguardanti la parte di progettazione in dettaglio e codifica;
- **Analisi dei Requisiti:** sono stati corretti alcuni scenari alternativi e inseriti come estensioni, come da suggerimento, e apportate piccole modifiche. Nel complesso il documento ha raggiunto un buon livello di maturità.
- **Piano di Progetto:** come suggerito, il documento è stato riorganizzato e nella forma e nei contenuti. Inoltre è stato incrementato con le parti relative alla progettazione;
- **Piano di Qualifica:** il documento ha subito profonde modifiche, è stato ristrutturato e riorganizzato. Sono state aggiunte le sezioni riguardanti le strategie e gli obiettivi di qualità. Il documento inoltre ha subito un incremento con l'aggiunta delle parti relative alla progettazione di dettaglio e codifica;
- **Glossario:** il documento ha subito piccole modifiche, ed è stato incrementato con termini nuovi.
- **Specifiche tecniche:** il documento ha subito profonde modifiche; come suggerito, sono stati rivisti e rifatti alcuni diagrammi ed è stato inserito il tracciamento mancante.