



# Piano di Qualifica

*Gruppo MINT – Progetto MaaS*

## Informazioni sul documento

<b>Versione</b>	3.0.0
<b>Redazione</b>	Fabiano Tavallini, Michael Ogbuachi, Navid Taha
<b>Verifica</b>	Tommaso Zagni, Thomas Fuser, Michael Ogbuachi
<b>Approvazione</b>	Enrico Canova
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo MINT

## Descrizione

Questo documento descrive le operazioni di verifica e validazione seguite dal gruppo MINT durante la realizzazione del progetto MaaS.



## Registro delle modifiche

Versione	Data	Collaboratori	Descrizione
3.0.0	18-05-2016	Enrico Canova (Responsabile)	Approvazione del documento.
2.2.0	18-05-2016	Tommaso Zagni, Michael Ogbuachi (Verificatori)	Verifica del documento.
2.1.0	17-05-2016	Tommaso Zagni, Thomas Fuser (Verificatori)	Verifica del documento.
2.0.5	16-05-2016	Navid Taha, Mi- chael Ogbuachi (Progettisti)	Sezione A5: stesura Test di Unità.
2.0.4	15-05-2016	Navid Taha, Fa- biano Tavallini (Progettisti)	Sezione A4: stesura Test di Integrazione.
2.0.3	14-05-2016	Fabiano Tavallini (Progettista)	Sezione A3: stesura Test di Sistema.
2.0.2	13-05-2016	Michael Ogbua- chi, Navid Taha (Progettisti)	Sezione A2: stesura Test di Validazione.
2.0.1	13-05-2016	Fabiano Tavallini (Progettista)	Sezione A1: stesura Livelli di Testing.
2.0.0	22-04-2016	Thomas Fuser, Fabiano Tavallini (Responsabili)	Approvazione del documento.
1.2.0	22-04-2016	Enrico Canova, Fabiano Tavallini, Michael Ogbuachi (Verificatori)	Verifica del documento.
1.1.0	21-04-2016	Enrico Cano- va, Navid Taha (Verificatori)	Verifica del documento.
1.0.3	20-04-2016	Thomas Fuser (Amministratore)	Appendice D: stesura avanzamento PDCA.
1.0.2	19-04-2016	Tommaso Za- gni, Navid Taha (Amministratori)	Sezione 2 spostata in Appendice C: Qualità.
1.0.1	18-04-2016	Tommaso Zagni (Amministratore)	Ridefinizione scheletro del documento.
1.0.0	27-03-2016	Michael Ogbuachi (Responsabile)	Approvazione del documento.
0.2.0	27-03-2016	Tommaso Zagni, Fabiano Tavallini (Verificatori)	Verifica finale del documento.



0.1.0	26-03-2016	Tommaso Zagni (Verificatore)	Verifica del documento fino a 3.7.
0.0.7	25-03-2016	Michael Ogbuachi, Fabiano Tavallini (Amministratori)	Appendice A: Stesura Resoconto delle attività di verifica.
0.0.6	24-03-2016	Fabiano Tavallini, Thomas Fuser (Amministratori)	Sezione 4: Stesura Gestione Amministrativa della Revisione.
0.0.5	22-03-2016	Michael Ogbuachi, Navid Taha (Amministratori)	Sezione 3.7.3: fine stesura. Completamento stesura Misure e Metriche.
0.0.4	21-03-2016	Michael Ogbuachi (Amministratore)	Inizio stesura Misure e Metriche Completate sezioni 3.7.1 e 3.7.2.
0.0.3	20-03-2016	Enrico Canova, Fabiano Tavallini (Amministratori)	Inizio stesura sezione 3. Completate le sezioni da 3.1 a 3.6.
0.0.2	19-03-2016	Fabiano Tavallini, Navid Taha (Amministratori)	Sezione 2: Stesura definizione obiettivi di qualità.
0.0.1	18-03-2016	Fabiano Tavallini (Amministratore)	Impostazione scheletro del documento. Sezione 1: Stesura Introduzione.



## Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo del documento . . . . .	6
1.2	Scopo del prodotto . . . . .	6
1.3	Glossario . . . . .	6
1.4	Riferimenti . . . . .	6
1.4.1	Normativi . . . . .	6
1.4.2	Informativi . . . . .	7
<b>2</b>	<b>Visione generale della strategia di verifica</b>	<b>7</b>
2.1	Definizione obiettivi di qualità . . . . .	7
2.1.1	Qualità di processo . . . . .	7
2.1.2	Qualità di prodotto . . . . .	8
2.2	Procedure di controllo di qualità . . . . .	8
2.2.1	Qualità di processo . . . . .	8
2.2.2	Qualità di prodotto . . . . .	8
2.3	Organizzazione . . . . .	8
2.3.1	Analisi . . . . .	9
2.3.2	Progettazione Architetturale . . . . .	9
2.3.3	Progettazione di Dettaglio e Codifica . . . . .	9
2.4	Strategia . . . . .	9
2.5	Responsabilità . . . . .	9
2.6	Risorse . . . . .	9
2.7	Misure e metriche . . . . .	10
2.7.1	Metriche per i documenti . . . . .	10
2.7.1.1	Gulpease . . . . .	10
2.7.2	Metriche per i processi . . . . .	11
2.7.2.1	Schedule Variance . . . . .	11
2.7.2.2	Budget Variance . . . . .	12
2.7.2.3	Produttività . . . . .	12
2.7.2.4	Impegno . . . . .	13
2.7.2.5	Modifiche . . . . .	13
2.7.2.6	Copertura dei test . . . . .	14
2.7.3	Metriche per il software . . . . .	14
2.7.3.1	Complessità ciclomatica . . . . .	14
2.7.3.2	Numero di metodi - NOM . . . . .	15
2.7.3.3	Variabili non utilizzate e non definite . . . . .	15
2.7.3.4	Numero parametri per metodo . . . . .	15
2.7.3.5	Halstead . . . . .	16
2.7.3.6	Maintainability index . . . . .	17
2.7.3.7	Use Case Points . . . . .	17
2.7.3.8	Statement Coverage . . . . .	18
2.7.3.9	Branch Coverage . . . . .	18
<b>3</b>	<b>Gestione amministrativa della revisione</b>	<b>18</b>
3.1	Comunicazione delle anomalie . . . . .	18
3.2	Procedure di controllo per la qualità di processo . . . . .	19



<b>Appendici</b>	<b>20</b>
<b>A Pianificazione dei test</b>	<b>21</b>
A.1 Livelli di testing . . . . .	21
A.2 Test di validazione . . . . .	22
A.3 Test di sistema . . . . .	29
A.4 Test di integrazione . . . . .	43
A.5 Test di unità . . . . .	45
<b>B Resoconto delle attività di verifica</b>	<b>83</b>
B.1 Riassunto delle attività di verifica . . . . .	83
B.1.1 Revisione dei Requisiti . . . . .	83
B.1.2 Revisione di Progettazione . . . . .	83
B.2 Dettaglio delle verifiche tramite analisi . . . . .	83
B.2.1 Analisi . . . . .	83
B.2.1.1 Documenti . . . . .	83
B.2.2 Progettazione . . . . .	83
B.2.2.1 Documenti . . . . .	83
B.3 Dettaglio dell'esito delle revisioni . . . . .	84
B.3.1 Revisione dei Requisiti . . . . .	84
<b>C Qualità</b>	<b>85</b>
C.1 Qualità di processo . . . . .	85
C.1.1 Standard ISO/IEC 15504 . . . . .	85
C.1.2 Ciclo di Deming . . . . .	86
C.2 Qualità di prodotto . . . . .	87
C.2.1 Standard ISO/IEC 9126 . . . . .	87
<b>D PDCA</b>	<b>89</b>
D.1 Revisione dei requisiti . . . . .	89
D.2 Revisione di progettazione . . . . .	89

## Elenco delle tabelle

A.1 Test di Validazione . . . . .	28
A.2 Test di Sistema . . . . .	42
A.3 Descrizione test d'Integrazione . . . . .	44
A.4 Test di Unità . . . . .	82
A.5 Esiti verifica documenti in fase di Analisi . . . . .	83
A.6 Esiti verifica documenti, Progettazione Architettuale . . . . .	84
A.7 Risultati metriche per i processi, Revisione di progettazione . . . . .	89
A.8 Risultati metriche per i processi, Revisione di progettazione . . . . .	90

## Elenco delle figure

1 Il ciclo di miglioramento dei processi . . . . .	19
2 V-Model per il testing software . . . . .	22



3	Sequenza d'integrazione delle componenti . . . . .	43
4	Continuous quality improvement with PDCA . . . . .	86



## 1 Introduzione

L'obiettivo primario è la *qualità<sub>G</sub>* del prodotto e dei suoi processi, ottenibile mediante una serie di controlli stabiliti inizialmente. L'assenza di queste verifiche, combinata ad un *team di sviluppo<sub>G</sub>* con più componenti senza particolari accortezze e competenze, porta al progressivo deterioramento del materiale prodotto, sia esso codice sorgente o documentazione.

Bisogna pertanto prevenire l'inserimento, all'interno del *repository<sub>G</sub>*, di materiale non congruo alle *Norme di Progetto v3.0.0* poiché si avvierebbe un graduale degrado della sua qualità.

### 1.1 Scopo del documento

Il Piano di Qualifica illustra la strategia di *verifica<sub>G</sub>* e *validazione<sub>G</sub>* che il gruppo MINT ha deciso di adottare per lo svolgimento del progetto. È necessario dimensionare la qualità dei prodotti e dei processi, operazione che non rientra nei normali ruoli di progettazione, bensì rappresenta una *funzione aziendale<sub>G</sub>*. Secondo le strategie riportate in questo documento il *Committente<sub>G</sub>* sarà in grado di valutare oggettivamente quanto è stato prodotto e disporrà di una solida base di verifica.

### 1.2 Scopo del prodotto

L'obiettivo preposto dal Committente per il progetto *MaaS<sub>G</sub>* (*MongoDB<sub>G</sub>* as an admin Service), è quello di costruire un servizio web che incorpora *MaaP<sub>G</sub>* (MongoDB as an admin Platform) e lo rende direttamente disponibile, attraverso il Web, per molteplici società. L'architettura di MaaS servirà gruppi di utenti rendendogli disponibile una condivisione dedicata di una sua istanza, includendo tutti i dati e le funzionalità in loro possesso.

### 1.3 Glossario

Ogni occorrenza di termini tecnici, di dominio e gli acronimi sono evidenziati in corsivo e marcati con la lettera G in pedice. I relativi significati sono riportati nel documento *Glossario v3.0.0*.

### 1.4 Riferimenti

Vengono elencati qui di seguito i riferimenti sui quali si basano il presente documento stesso e l'organizzazione delle attività di verifica e validazione.

#### 1.4.1 Normativi

- **Norme di progetto:** *Norme di Progetto v3.0.0*;
- **Capitolato d'appalto C4:** RedBabel, MaaS <http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C4.pdf>;
- **Standard ISO/IEC 15504:** [http://en.wikipedia.org/wiki/ISO/IEC\\_15504](http://en.wikipedia.org/wiki/ISO/IEC_15504);

- **Standard ISO/IEC 9126:** [http://en.wikipedia.org/wiki/ISO/IEC\\_9126](http://en.wikipedia.org/wiki/ISO/IEC_9126);
- **Standard IEEE 610.12-90:** [https://cow.ceng.metu.edu.tr/Courses/download\\_courseFile.php?id=2677](https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=2677).

#### 1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v3.0.0*;
- **SWEBOK v3:** capitolo 10;
- **Slides del corso di Ingegneria del Software mod. A:** Qualità del software <http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/L08.pdf>;
- **Slides del corso di Ingegneria del Software mod. A:** Qualità del *processo<sub>G</sub>* <http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/L09.pdf>;
- **Software Engineering 9th - I. Sommerville (Pearson, 2011):** capitoli 24 e 26;
- **Metriche del software G - Ercole F. Colonese:** [http://www.colonese.it/00-Manuali\\_Pubblicatii/08-Metriche%20del%20software\\_v1.0.pdf](http://www.colonese.it/00-Manuali_Pubblicatii/08-Metriche%20del%20software_v1.0.pdf);
- **Ciclo di Deming:** [https://it.wikipedia.org/wiki/Ciclo\\_di\\_Deming](https://it.wikipedia.org/wiki/Ciclo_di_Deming);
- **Indice Gulpease:** [https://it.wikipedia.org/wiki/Indice\\_Gulpease](https://it.wikipedia.org/wiki/Indice_Gulpease).

## 2 Visione generale della strategia di verifica

La strategia adottata ha lo scopo di automatizzare il lavoro di verifica; tale scelta necessita l'uso di tool adeguatamente configurati. Lo scopo è ottenere un riscontro affidabile e numericamente trattabile che consenta di assicurare il grado di qualità predeterminato. L'aspettativa è la riduzione del lavoro manuale permettendo così una validazione semplificata.

### 2.1 Definizione obiettivi di qualità

#### 2.1.1 Qualità di processo

La qualità del processo è un fattore determinante per la qualità del prodotto. Si è deciso di perseguire la qualità servendosi di due modelli:

- *SPICE<sub>G</sub>*<sup>1</sup>, definito nello standard ISO/IEC 15504, ai fini di una valutazione oggettiva dei processi, per darne un giudizio di maturità e per individuare azioni migliorative;
- *PDCA<sub>G</sub>*<sup>2</sup>, per un miglioramento continuo della qualità. In particolare permette un controllo delle attività di processo ripetibili e misurabili garantendo la manutenzione dei processi stessi incrementandone la qualità.

---

<sup>1</sup>Si veda appendice C.1 per approfondimenti

<sup>2</sup>Si veda appendice C.1.2 per approfondimenti





### 2.1.2 Qualità di prodotto

Oltre alla qualità di processo, sono necessari degli obiettivi rivolti direttamente alla qualità del prodotto per massimizzare l'efficacia. A tal fine, lo standard ISO/IEC 9126<sup>3</sup> classifica la qualità del software e definisce delle metriche per la sua misurazione.

## 2.2 Procedure di controllo di qualità

### 2.2.1 Qualità di processo

Le linee guida per la gestione della qualità di processo seguono il modello PDCA descrivendo come devono essere attuate le procedure di controllo:

- Pianificazione dettagliata;
- Monitoraggio delle attività pianificate;
- Definizione delle risorse necessarie al conseguimento degli obiettivi;
- Utilizzo di metriche per verificare il miglioramento della qualità dei processi.

All'interno del *Piano di Progetto v3.0.0* è descritta in dettaglio la pianificazione di queste attività per il miglioramento continuo dei processi, ottenibile in modo indiretto con la costante analisi della qualità di prodotto. Un prodotto di bassa qualità indica indubbiamente che a monte vi sta un processo migliorabile. Inoltre, per quantificare la qualità dei processi, si possono usare delle metriche. Quelle adottate sono descritte nella sezione 2.7.

### 2.2.2 Qualità di prodotto

Grazie ai sequenti processi verrà garantito il controllo di qualità del prodotto:

- **Software Quality Assurance ( $SQA_c$ )**: l'insieme delle attività realizzate al fine di garantire il raggiungimento degli obiettivi di qualità; è importante che tale processo sia preventivo e non correttivo;
- **Verifica**: assicura che l'esecuzione delle attività dei processi svolti non introduca errori nel prodotto. Durante l'intera durata del progetto verranno svolte attività di verifica sugli output dei processi, accertando che esso sia corretto, completo e rispetti regole, convenzioni e procedure;
- **Validazione**: la conferma oggettiva che assicura che i prodotti finali soddisfino i requisiti e le aspettative attese.

## 2.3 Organizzazione

Viene verificata la qualità dei singoli processi e dei loro output. La verifica degli output dei periodi descritti nel *Piano di Progetto v3.0.0* è programmata in attività mirate.

---

<sup>3</sup>Si veda appendice C.2 per approfondimenti



### 2.3.1 Analisi

Periodo che controlla il rispetto dei processi e della documentazione prodotta rispetto le *Norme di Progetto v3.0.0*. Sarà inoltre verificata la corrispondenza tra requisiti e casi d'uso.

### 2.3.2 Progettazione Architettuale

Periodo di verifica dei processi incrementali relativi all'analisi e ai nuovi documenti di progettazione. Verifica inoltre che i test siano adeguatamente pianificati come descritto nel *Piano di Progetto v3.0.0* ed eseguiti secondo quanto descritto nelle *Norme di Progetto v3.0.0*.

### 2.3.3 Progettazione di Dettaglio e Codifica

Periodo di verifica dei processi incrementali relativi alla progettazione assieme alla verifica delle attività di codifica tramite tecniche di analisi statica e dinamica.

Il Diario delle modifiche viene incluso in ogni documento mostrando così lo storico del documento.

## 2.4 Strategia

Il *Piano di Progetto v3.0.0* fissa una serie di scadenze improrogabili, risulta necessario definire con chiarezza una strategia di qualifica efficace. Gli incrementi sulla documentazione o sul codice possono essere di natura programmata, quindi prefissati nel calendario, oppure possono insorgere come inaspettati. In questo caso sarà necessario programmare le dovute modifiche; è questo il caso di *bug<sub>c</sub>* o errori (vedi paragrafo 3.1). La qualità di ogni incremento è basata sul fatto che la struttura di qualifica garantisce il rispetto delle *Norme di Progetto v3.0.0*. Tale lavoro verrà svolto con l'aiuto di automatismi che segnaleranno le problematiche rilevate in modo da permettere una rapida correzione. L'utilizzo di software apposito permette di eseguire controlli mirati senza consumare risorse umane. L'implementazione di tali controlli viene descritta nelle *Norme di Progetto v3.0.0*.

## 2.5 Responsabilità

La responsabilità delle verifiche è attribuita al Responsabile di progetto e ai Verificatori. All'interno del *Piano di Progetto v3.0.0* sono definiti i compiti e le modalità di attuazione.

## 2.6 Risorse

La qualifica dei processi, essendo anch'essa un processo, consuma delle risorse distinguibili in due categorie:

- **Umane:** le figure coinvolte sono il Responsabile di progetto e il Verificatore. I processi da loro effettuati consumano ore di produttività contabilizzate e schedate secondo il *Piano di Progetto v3.0.0*. Le ore di produttività sono fissate dalle regole di progetto (<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/PD01b.html>) in un minimo di 85 e un massimo di 105 ore individuali. Il *Piano di Progetto v3.0.0* determina la distribuzione



di tali quote orarie con la relativa retribuzione. Ai fini della qualifica si potrà parlare di ore di produttività, tralasciandone l'aspetto economico, in quanto non rientra nel dominio del documento succitato;

- **Tecnologiche:** riguardano i mezzi utilizzati per gli automatismi, la qualità e la loro gestione. Trattandosi esclusivamente di mezzi informatici, vengono consumate unità di calcolo considerate a costo nullo. Tale considerazione si basa sul fatto che tutti i tipi di elaborazioni informatiche sono svolte su mezzi per i quali non è richiesto né un contributo economico, né un quantitativo temporale abbastanza consistente da poter essere considerato degno di nota.

Le modalità del loro impiego sono descritte all'interno del documento *Norme di Progetto v3.0.0*.

## 2.7 Misure e metriche

Per monitorare l'andamento dei processi e la qualità del prodotto vengono adottate delle metriche che rendono misurabili e valutabili i processi, i documenti ed il software prodotto.

### 2.7.1 Metriche per i documenti

La leggibilità dei documenti ne garantisce la qualità. Si utilizza un indice per misurare la leggibilità dei testi in lingua italiana.

#### 2.7.1.1 Gulpease

L'indice Gulpease è un indice per la leggibilità di un testo tarato sulla lingua italiana. Questo indice ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. Permette di misurare la complessità dello stile di un documento. In questo calcolo sono considerate due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

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

Il range dei risultati varia tra 0 e 100, dove 100 indica la leggibilità massima e 0 quella minima. In generale i documenti con:

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

#### Parametri utilizzati

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



### 2.7.2 Metriche per i processi

Tali metriche monitorano e prevedono l'andamento delle principali variabili critiche del progetto: tempi e costi. Sono utilizzate metriche di tipo consultivo le quali consentono un riscontro immediato sullo stato attuale del progetto; ad ogni incremento verranno valutati tali indici e, se necessario, verranno stabiliti opportuni provvedimenti da parte del Responsabile di progetto.

#### 2.7.2.1 Schedule Variance

Permette di calcolare le tempistiche rispetto la schedulazione delle attività pianificate alla data corrente. È un indicatore di efficacia soprattutto nei confronti del Cliente.

$$SV = BCWP - BCWS$$

Dove:

- **BCWP**: indica il valore delle attività realizzate alla data corrente;
- **BCWS**: indica il costo pianificato per realizzare le attività di progetto alla data corrente.

Quindi con:

- **SV>0**: il lavoro prodotto è in anticipo rispetto quanto pianificato;
- **SV<0**: il lavoro è in ritardo;
- **SV=0**: il lavoro è in linea con quanto stabilito.

### 2.7.2.2 Budget Variance

Permette di calcolare i costi rispetto alla data corrente. È un indicatore che ha un valore unicamente contabile e finanziario.

$$BV = BCWS - ACWP$$

Dove:

- **BCWS**: indica il costo pianificato per realizzare le attività di progetto alla data corrente;
- **ACWP**: indica il costo effettivamente sostenuto per realizzare le attività di progetto alla data corrente.

Quindi:

- **BV>0**: il budget speso è minore rispetto quanto pianificato;
- **BV<0**: il budget è maggiore di quanto pianificato in ritardo;
- **BV=0**: il budget speso è in linea con quanto stabilito.

### 2.7.2.3 Produttività

#### Produttività di documentazione

Indica la produttività media di documentazione delle risorse impiegate, valutando quindi le persone coinvolte durante i diversi stadi del progetto.

$$\text{Produttività di documentazione} = \text{Parole} / \text{Ore persona}$$

Dove:

- **Parole**: indica il numero di parole presente nei documenti;
- **Ore persona**: indica il numero di ore produttive dei componenti del gruppo.

#### Parametri utilizzati

- **Range-ottimale**:  $[\geq 100]$ .



### Produttività di test

Indica la produttività media dei test realizzati.

$$\text{Produttività di test} = \text{Numero di test} / \text{Ore persona}$$

Dove:

- **Numero di test:** indica il numero di test eseguiti;
- **Ore persona:** indica il numero di ore produttive dei componenti del gruppo.

### Produttività di codifica

Indica la produttività media delle attività di codifica.

$$\text{Produttività di codifica} = \text{LOCs} / \text{Ore persona}$$

Dove:

- **LOCs:** indica il numero di linee di codice prodotte;
- **Ore persona:** indica il numero di ore produttive dei componenti del gruppo.

#### 2.7.2.4 Impegno

Indica l'impegno richiesto dal gruppo per la realizzazione del progetto.

$$\text{Impegno} = \text{Dimensione} / \text{Produttività}$$

Dove:

- **Dimensione:** indica il tempo produttivo impiegato;
- **Produttività:** indica la media delle produttività totali (di documentazione, di test, di codifica).

### Parametri utilizzati

- **Range-ottimale:**  $[\geq 0,6]$ .

#### 2.7.2.5 Modifiche

Indica il numero di modifiche approvate dal responsabile. Le modifiche possono riguardare requisiti, funzionalità, codice e documenti.

$$\text{Modifiche} = \text{Numero di modifiche}$$

Dove:

- **Numero di modifiche:** indica le issue etichettate come “richiesta di modifica” e “approvate”.

**Parametri utilizzati**

- **Range-accettazione:** [0 - 20];
- **Range-ottimale:** [0 - 10].

**2.7.2.6 Copertura dei test**

Indica la percentuale di casi coperti da test eseguiti.

*Copertura del test = Numero di funzioni testate \* 100 / Numero totale di funzioni disponibili*

**Parametri utilizzati**

- **Range-accettazione:** [70 - 100];
- **Range-ottimale:** [80 - 100].

**2.7.3 Metriche per il software**

La prima *release<sub>G</sub>* di *Node.js<sub>G</sub>* risale a maggio 2009. È riscontrata una forte differenza tra le metriche disponibili per l'analisi statica rispetto a quelle per i linguaggi meno recenti. Nessun membro del gruppo ha conoscenza di tale linguaggio e delle sue particolarità, soprattutto riguardo all'aspetto *funzionale<sub>G</sub>*. Le differenze con i linguaggi studiati nel percorso universitario portano a difficoltà nell'individuare le metriche non incentrate sulla visione ad oggetti del codice. Infine si è osservata l'assenza di strumenti per la misurazione di metriche tradizionali come la coesione e l'instabilità dei *package<sub>G</sub>*. Di seguito vengono elencate le metriche per il software prodotto.

**2.7.3.1 Complessità ciclomatica**

La complessità ciclomatica è una metrica software che indica la complessità di un programma misurando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso. Nel grafo sopraccitato i nodi corrispondono a gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo di istruzioni può essere eseguito immediatamente dopo il primo gruppo. Tale indice può essere applicato indistintamente a singole funzioni, *moduli<sub>G</sub>*, metodi o package di un programma. Si vuole utilizzare tale metrica per limitare la complessità durante le attività di sviluppo del prodotto software. Può rivelarsi utile durante il testing per determinare il numero di casi di test necessari, infatti l'indice di complessità è un limite superiore al numero di test necessari per raggiungere il coverage completo del modulo testato. Inoltre, uno studio ha mostrato forti corrispondenze tra le metriche di complessità e il livello di coesione nei package presi in esame<sup>4</sup>.

---

<sup>4</sup>Stein, C., G. Cox and L. Etzkorn, 2005. Exploring the Relationship between Cohesion and Complexity. J. Comput. Sci., 1: 137-144.

**Parametri utilizzati**

- **Range-accettazione:**  $[0 - 25]$ ;
- **Range-ottimale:**  $[0 - 10]^5$ .

**2.7.3.2 Numero di metodi - NOM**

Il Number of methods è una metrica usata per calcolare la media delle occorrenze dei metodi per package. Un package non dovrebbe contenere un numero eccessivo di metodi. Valori superiori al range ottimale massimo potrebbero indicare una necessità di maggiore decomposizione del package.

**Parametri utilizzati**

- **Range-accettazione:**  $[3 - 10]$ ;
- **Range-ottimale:**  $[3 - 7]$ .

**2.7.3.3 Variabili non utilizzate e non definite**

La presenza di variabili non utilizzate viene considerata *pollution<sub>G</sub>*, pertanto non viene tollerata. Tali occorrenze vengono rilevate analizzando l'*Abstract syntax tree<sub>G</sub>* (AST) eseguendo una cross-reference tra le variabili dichiarate e quelle inizializzate. Per sua natura, *Javascript<sub>G</sub>* non blocca l'insorgenza di tali occorrenze, pertanto si rischia di dichiarare una variabile e poi utilizzarne una con nome leggermente diverso, oppure semplicemente dichiarare una variabile che in seguito non verrà mai utilizzata.

**Parametri utilizzati**

- **Range-accettazione:**  $[0 - 0]$ ;
- **Range-ottimale:**  $[0 - 0]$ .

**2.7.3.4 Numero parametri per metodo**

Un numero elevato di parametri per un metodo potrebbe evidenziare un metodo troppo complesso.

Non c'è una regola forte per il numero di parametri possibili in un metodo o costruttore, citando Robert Martin, in Clean Code<sup>6</sup>:

*"The ideal number of arguments for a function is zero (niladic). Next comes one (monadic), followed closely by two (dyadic). Three arguments (triadic) should be avoided where possible. More than three (polyadic) requires very special justification – and then shouldn't be used anyway."*

e Steve McConnell, in Code Complete<sup>7</sup>:

*"limit the number of a routine's parameters to about seven, seven is a magic number for people's*

---

<sup>5</sup>McCabe (dicembre 1976). A Complexity Measure. IEEE Transactions on Software Engineering: 308–320.

<sup>6</sup>Robert Martin, Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall (2008)

<sup>7</sup>Steve McConnell, Code Complete: A Practical Handbook of Software Construction. Microsoft Press (2004)



*comprehension*”

Vengono quindi seguite le linee guida dei seguenti parametri.

#### Parametri utilizzati

- **Range-accettazione:** [0 - 8];
- **Range-ottimale:** [0 - 4].

#### 2.7.3.5 Halstead

La metrica di *Halstead<sub>G</sub>* oltre ad essere un indice di complessità, permette di identificare le proprietà misurabili del software e le relative relazioni. Si basa sull’osservazione che una metrica dovrebbe valutare l’implementazione di un algoritmo in linguaggi differenti ed essere indipendente dall’esecuzione su una specifica piattaforma.

Sono identificati i seguenti dati all’interno di un problema:

- **$n_1$**  : indica il numero di operatori distinti;
- **$n_2$** : indica il numero di operandi distinti;
- **$N_1$**  : indica il numero totale di operatori;
- **$N_2$**  : indica il numero totale di operandi.

Da cui si ottiene:

- **$n = n_1 + n_2$** : vocabolario della funzione;
- **$N = N_1 + N_2$** : lunghezza della funzione.

Data la scarsa disponibilità nella rete di valori di riferimento, i range specificati sono frutto di un confronto tra il *report<sub>G</sub>* sulla complessità di una libreria *open source<sub>G</sub>* presa come esempio (<https://github.com/philbooth/complexity-report/blob/master/EXAMPLE.md>) e i valori dichiarati in <http://www.mccabe.com/pdf/McCabeIQMetrics.pdf>. Questi valori vengono dichiarati momentanei (RR) e saranno da rivalutare sia considerando altre fonti, sia considerando i valori rilevati in parti del codice che il gruppo considera come riferimento.

#### Halstead difficulty per-function

Il livello di difficoltà di una funzione misura la propensione all’errore ed è proporzionale al numero di operatori presenti.

$$D = \left(\frac{n_1}{2}\right) * \left(\frac{N_2}{n_2}\right)$$

#### Parametri utilizzati

- **Range-accettazione:** [0 - 30];
- **Range-ottimale:** [0 - 15].



### Halstead volume per-function

Il volume descrive la dimensione dell'implementazione di un algoritmo e si basa sul numero di operazioni eseguite e sugli operandi di una funzione. Il volume di una function senza parametri composta da una sola linea è 20, mentre un indice superiore a 1000 indica che probabilmente la funzione esegue troppe operazioni.

$$V = N * \log_2 n$$

#### Parametri utilizzati

- **Range-accettazione:** [20 - 1500];
- **Range-ottimale:** [20 - 1000].

### Halstead effort per-function

Lo sforzo per implementare o comprendere il significato di una funzione è proporzionale al volume a al suo livello di difficoltà.

$$E = V * D$$

#### Parametri utilizzati

- **Range-accettazione:** [0 - 400];
- **Range-ottimale:** [0 - 300].

### 2.7.3.6 Maintainability index

Questa metrica<sup>8</sup> è una scala logaritmica da  $-\infty$  a 171, calcolata sulla base delle linee di codice logiche, della complessità ciclomatica e dall'indice Halstead effort. Un valore alto indica una maggiore manutenibilità.

#### Parametri utilizzati

- **Range-accettazione:** [ $>70$ ];
- **Range-ottimale:** [ $>90$ ].

### 2.7.3.7 Use Case Points

Gli Use Case Points (UCP)<sup>9</sup> stimano quanto sforzo è necessario per sviluppare il prodotto basandosi su quanto lavoro il software deve svolgere.

Al fine di stimare un costo in ore-uomo di sviluppo, tale tecnica valuta i seguenti fattori:

- **Fattori tecnici dell'implementazione:** principalmente i requisiti non funzionali del sistema;

---

<sup>8</sup>Definita nel 1991 da Paul Oman e Jack Hagemeister alla University of Idaho.

<sup>9</sup>Definiti da Gustav Karner della Rational Software Corporation a metà del 1990.

- **Fattori ambientali:** per lo più caratterizzati dalla composizione del team;
- **Quantità e complessità degli use case:** costituiscono il numero degli use case e il numero di transizioni all'interno degli use case;
- **Quantità e complessità degli attori:** il numero di attori e come si interfacciano al sistema.

#### 2.7.3.8 Statement Coverage

Permette di calcolare quante linee di comando di ciascun modulo delle unità sono eseguite almeno una volta nell'esecuzione dei test. Tale metrica è espressa in percentuale.

##### Parametri utilizzati

- **Range-accettazione:** [70 - 100];
- **Range-ottimale:** [85 - 100].

#### 2.7.3.9 Branch Coverage

Permette di calcolare quanti rami della logica di flusso sono attraversati almeno una volta durante l'esecuzione dei test. Tale metrica è espressa in percentuale.

##### Parametri utilizzati

- **Range-accettazione:** [70 - 100];
- **Range-ottimale:** [85 - 100].

## 3 Gestione amministrativa della revisione

### 3.1 Comunicazione delle anomalie

Identificare le anomalie permette la correzione dei difetti ricercati dal processo di *Software Quality Management*<sub>e</sub> e informa il Responsabile di progetto sullo stato del prodotto. Analizzare e catalogare le anomalie è utile per discutere, durante revisioni e riunioni, su quali modifiche e correzioni applicare e con quale priorità. Di seguito è presente la lista delle definizioni di anomalie (IEEE 610.12-90) adottate dal gruppo:

- **Error:** differenza riscontrata tra il risultato di una computazione e il valore teorico atteso (e.g. uscita dal range di accettazione degli indici di misurazione);
- **Fault:** un passo, un processo o un dato definito in modo erraneo (e.g. violazioni di norme tipografiche da parte di un documento). Corrisponde a quanto viene definito come bug;
- **Failure:** il risultato di un fault (e.g. incongruenza del prodotto con funzionalità indicate nell'analisi dei requisiti, incongruenza del codice con il design del prodotto);

- **Mistake:** azione umana che produce un risultato errato (e.g. anomalie nel repository).

La distinzione delle anomalie consente di impostare le metriche per valutarne l'andamento e in alcuni casi predirlo, in particolare è stata scelta la metrica che conta il numero di bug per lines of code. Il gruppo utilizzerà un  $SCR_G$  (Software Change Request) individuato nelle *Norme di Progetto v3.0.0*.

### 3.2 Procedure di controllo per la qualità di processo

Le procedure di controllo per la qualità di processo hanno il fine di migliorare la qualità del prodotto e diminuire i costi e tempi di sviluppo. Esistono due approcci principali:

- **A maturità di processo:** riflette le buone pratiche di management e tecniche di sviluppo. L'obiettivo primario è la qualità del prodotto e la prevedibilità dei processi;
- **Agile:** sviluppo iterativo senza l'overhead della documentazione e di tutti gli aspetti pre-determinabili. Ha come caratteristica la responsività ai cambiamenti dei requisiti cliente e uno sviluppo rapido.

Il team adotterà il primo approccio, essendo più adatto ad un gruppo inesperto. Con una visione proattiva si cerca di avere maggior controllo e previsione sulle attività da svolgere. Questa viene anche indicata come *best practice<sub>G</sub>* per gruppi poco esperti.

Il processo con maggiore influenza sulla qualità del sistema non è quello di sviluppo ma quello di progettazione. È qui che le capacità e le esperienze dei singoli danno un contributo decisivo. Il miglioramento dei processi è un processo ciclico composto da tre sotto-processi:

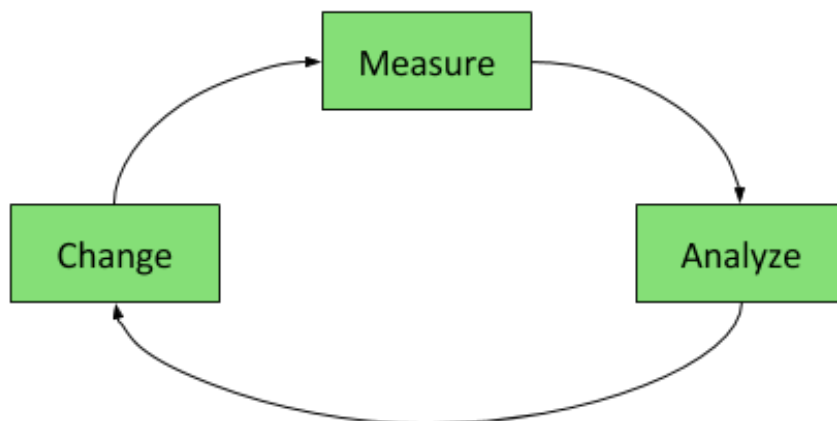


Figura 1: Il ciclo di miglioramento dei processi

- **Misurazione del processo:** misura gli attributi del progetto, punta ad allineare gli obiettivi con le misurazioni effettuate. Questo forma una *baseline<sub>G</sub>* che aiuta a capire se i miglioramenti hanno avuto effetto;
- **Analisi del processo:** vengono identificate le problematiche ed i colli di bottiglia dei processi;



- **Modifiche del processo:** i cambiamenti vengono proposti in risposta alle problematiche riscontrate.

Il team procederà nel seguente modo:

- Nella sezione Dettaglio delle verifiche tramite analisi (B.2) verranno inserite le misurazioni rilevate sulle le metriche descritte in Misure e Metriche (2.7);
- L'analisi viene effettuata i giorni precedenti alle consegne previste dal committente. Il Resoconto delle attività di verifica (B) contiene l'analisi del processo e le relative considerazioni comprendenti le problematiche riscontrate;
- Le modifiche al processo vengono attuate all'inizio del processo incrementale successivo. Queste attività sono programmate nel *Piano di Progetto v3.0.0*.



## A Pianificazione dei test

Si vuole adottare una strategia di verifica del software tramite test opportunamente predeterminati, garantendo almeno un test per ogni requisito. Essi sono l'applicazione delle tecniche di verifica dinamica introdotte nelle *Norme di Progetto v3.0.0*. Tali attività, oltre a richiedere l'esecuzione del programma, devono poter essere ripetibili, ossia tramite delle specifiche su come riprodurre i test vogliamo che il loro output sia deterministico. È importante che i test di unità vengano svolti in parallelo, dando precedenza alle unità che producono risultati utili alla comprensione del loro funzionamento integrato. L'ambiente di testing deve soddisfare tale obiettivo. L'attività di test deve produrre un  $\log_G$  che specifica quando e chi ha eseguito il test e con quali input; l'insorgenza di  $failure_G$  deve essere tracciata e catalogata.

### A.1 Livelli di testing

Il testing del software viene suddiviso in livelli differenti e si concretizzano in un'esecuzione bottom-up che avanza sequenzialmente alle attività di codifica e di validazione. I test che si andranno ad applicare sono di cinque tipi:

- **Test di Validazione (TV):** viene verificato che il prodotto soddisfi quanto richiesto dal proponente individuando delle macro azioni da eseguire sul sistema che un normale utente svolge comunemente;
- **Test di Sistema (TS):** sono test relativi al comportamento dell'intero sistema ossia viene verificato che la sua architettura generale funziona complessivamente bene;
- **Test di Integrazione (TI):** vengono verificate le componenti del sistema contenute nella *Specifica Tecnica v0.0.0*, ossia viene verificato che i  $package_G$  siano funzionanti e in grado di funzionare nel loro insieme;
- **Test di Unità (TU):** viene testata ogni unità, ossia la più piccola parte di lavoro assegnabile ad un programmatore. In questo progetto una unità corrisponde ad una `function` o a un `method`;
- **Test di Regression (TR):** possono essere test di tutte le tipologie succitate che devono mostrare il funzionamento del prodotto a seguito di una modifica.

La specifica delle ultime due tipologie viene riservata alla prossima revisione.

La figura 2 illustra come i test elencati vengono distribuiti durante il ciclo di sviluppo del prodotto.

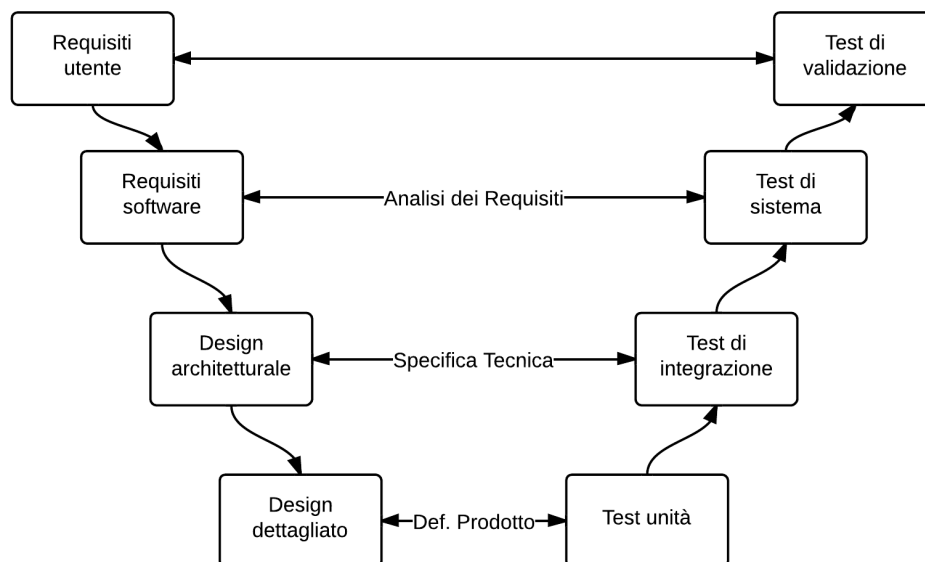


Figura 2: V-Model per il testing software

## A.2 Test di validazione

In questa sezione vengono elencati i test di validazione per verificare che il prodotto sia conforme alle attese. I test si svolgono seguendo e verificando tutti passi di cui si compongono.



Test di Validazione	Descrizione	Stato
TV-RS1O 1	L'utente non autenticato intende autenticarsi inserendo i dati necessari in un'apposita form nella pagina di autenticazione. All'utente è richiesto di: <ul style="list-style-type: none"><li>• Raggiungere la pagina di autenticazione;</li><li>• Inserire l'email e la password;</li><li>• Procedere con l'autenticazione.</li></ul>	Non eseguito
TV-RS1O 2.1	L'utente non autenticato intende registrarsi compilando i campi della form nella pagina di registrazione. All'utente è richiesto di: <ul style="list-style-type: none"><li>• Raggiungere la pagina di registrazione;</li><li>• Inserire l'email nel campo apposito;</li><li>• Inserire la password conforme ai vincoli richiesti;</li><li>• Inserire la conferma della password;</li><li>• Inserire il nome dell'azienda che desidera creare;</li><li>• Procedere con la registrazione.</li></ul>	Non eseguito
TV-RS1O 2.2	L'utente non autenticato intende unirsi ad un'azienda già esistente, che l'ha invitato tramite email. All'utente è richiesto di: <ul style="list-style-type: none"><li>• Raggiungere la pagina di registrazione grazie al link ricevuto per email;</li><li>• Inserire la password conforme ai vincoli richiesti;</li><li>• Inserire la conferma della password;</li><li>• Procedere con la registrazione.</li></ul>	Non eseguito





TV-RS1D 3	<p>L'utente autenticato è intenzionato a modificare i dati del proprio account mediante una pagina web dedicata. All'utente è richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Aprire la pagina relativa alle modifiche dell'account;</li><li>• Decidere se modificare il proprio avatar;</li><li>• Decidere se modificare la propria password;</li><li>• Decidere se modificare i propri dati anagrafici quali nome, cognome e data di nascita;</li><li>• Decidere se modificare il proprio sesso;</li><li>• Rendere effettive le modifiche.</li></ul>	Non eseguito
TV-RS10 4	<p>Il Proprietario o l'Amministratore è intenzionato a connettere o disconnettere un database oppure modificarne i permessi mediante una pagina web dedicata. È richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Aprire la pagina relativa alla gestione database;</li><li>• Decidere se connettere un database inserendo il nome del nuovo database;</li><li>• Decidere se disconnettere un database;</li><li>• Decidere se modificare i permessi di un database ai membri dell'azienda;</li><li>• Decidere se cercare un database appartenente all'azienda;</li><li>• Rendere effettive le modifiche.</li></ul>	Non eseguito



TV-RS10 5	<p>L'utente autenticato è intenzionato ad eseguire operazioni di lettura, scrittura e modifica dei permessi in ambito dei DSLIS, mediate l'utilizzo di un editor testuale e un'interfaccia che visualizza la lista delle definizioni DSL. All'utente è richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Aprire l'editor di testo;</li><li>• Decidere se creare un nuovo DSLIS o modificare quelli già esistenti. É possibile definire una Dashboard, una Collection, un Document e una Cell;</li><li>• Decidere se eseguire una ricerca dei DSLIS sfoltendo i risultati della lista;</li><li>• Decidere se eseguire un DSLIS;</li><li>• Decidere se eseguire una Action sui dati visualizzati da un DSLIS;</li><li>• Decidere se esportare un DSLIS;</li><li>• Decidere se importare un DSLIS;</li><li>• Decidere se eliminare i DSLIS già esistenti;</li><li>• Decidere se visualizzare il codice di un DSLIS;</li><li>• Decidere se modificare i permessi di lettura, scrittura o esecuzione associati ad un DSLIS.</li></ul>	Non eseguito
TV-RS10 6	<p>Il Proprietario o l'Amministratore è intenzionato a manipolare gli account degli altri utenti dell'azienda. É richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato come Proprietario o Amministratore;</li><li>• Aprire la pagina relativa alla gestione account dei membri dell'azienda;</li><li>• Decidere se rimuovere un utente del tipo Amministratore, Membro o Ospite;</li><li>• Aggiungere un nuovo membro all'azienda, invitandolo tramite email e affidandogli un determinato ruolo;</li><li>• Decidere di modificare il ruolo di un utente promuovendolo o facendolo retrocedere di ruolo;</li><li>• Rendere effettive le modifiche.</li></ul>	Non eseguito



TV-RS10 7	L'utente autenticato è intenzionato a ad eseguire il logout dal sistema. È richiesto di: <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Eseguire il logout tramite l'apposito tasto presente in ogni pagina di MaaS.</li></ul>	Non eseguito
TV-RS10 8	L'utente non autenticato è intenzionato a recuperare la propria password, ripristinandola. È richiesto di: <ul style="list-style-type: none"><li>• Recarsi alla pagina di recupero password;</li><li>• Inserire la propria email e richiedere il ripristino;</li><li>• Seguire il link ricevuto nell'email inviata dal sistema;</li><li>• Inserire la nuova password nella pagina raggiunta tramite il link. Essa dovrà rispettare i requisiti ed essere inserita una seconda volta per conferma;</li><li>• Confermare i dati inseriti.</li></ul>	Non eseguito
TV-RS10 9	L'utente autenticato è intenzionato a modificare le proprie preferenze. È richiesto di: <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Decidere se modificare la Dashboard attiva tra quelle disponibili;</li><li>• Decidere se effettuare una ricerca tra Dashboard disponibili;</li><li>• Rendere effettive le modifiche.</li></ul>	Non eseguito
TV-RS10 10	L'utente autenticato è intenzionato a visualizzare la pagina relativa alla propria Dashboard. È richiesto di: <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Recarsi alla pagina di visualizzazione della Dashboard attiva.</li></ul>	Non eseguito
TV-RS10 12	L'utente autenticato, è intenzionato a cercare un altro utente iscritto al sistema. È richiesto di: <ul style="list-style-type: none"><li>• Essere autenticato;</li><li>• Effettuare una ricerca inserendo il nome o la mail dell'utente desiderato.</li></ul>	Non eseguito



TV-RS10 13.1	<p>Il Super-Amministratore è intenzionato a gestire le aziende contenute nel database interno del sistema. È richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato come Super-Amministratore;</li><li>• Recarsi alla pagina di gestione delle aziende;</li><li>• Decidere se ricercare un'azienda specifica;</li><li>• Decidere se modificare il nome di un'azienda;</li><li>• Decidere se eliminare un'azienda;</li></ul>	Non eseguito
TV-RS10 13.2	<p>Il Super-Amministratore è intenzionato a gestire gli utenti di un'azienda contenuti nel database interno del sistema. È richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato come Super-Amministratore;</li><li>• Recarsi alla pagina di gestione degli utenti di un'azienda;</li><li>• Decidere se modificare il nome di un utente;</li><li>• Decidere se modificare la password di un utente;</li><li>• Decidere se eliminare un utente;</li></ul>	Non eseguito
TV-RS10 14	<p>Il Super-Amministratore è intenzionato a impersonificare un utente. È richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato come Super-Amministratore;</li><li>• Recarsi alla pagina di visualizzazione degli utenti di un'azienda;</li><li>• Effettuare una ricerca sulla lista selezionando l'utente da impersonificare.</li></ul>	Non eseguito
TV-RS10 15	<p>Il Super-Amministratore è intenzionato ad autenticarsi mediante un'apposita pagina web. È richiesto di:</p> <ul style="list-style-type: none"><li>• Non essere autenticato;</li><li>• Recarsi alla pagina autenticazione del Super Amministratore;</li><li>• Inserire l'email e la password;</li><li>• Procedere con l'autenticazione.</li></ul>	Non eseguito



TV-RS10 16	<p>Il Super-Amministratore è intenzionato ad eseguire il logout. É richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato come Super-Amministratore;</li><li>• Selezionare l'apposito pulsante di logout presente in tutte le pagine.</li></ul>	Non eseguito
TV-RS10 18	<p>Il Super-Amministratore è intenzionato a cercare un utente. É richiesto di:</p> <ul style="list-style-type: none"><li>• Essere autenticato come Super-Amministratore;</li><li>• Effettuare una ricerca inserendo il nome o l'email dell'utente desiderato.</li></ul>	Non eseguito

Tabella A.1: Test di Validazione



### **A.3 Test di sistema**

Vengono qui descritti i test di sistema che andranno a verificare il funzionamento complessivo delle componenti.



Test di Sistema	Descrizione	Stato
TS-RS1O 1.1	Verificare che durante l'autenticazione di un utente l'indirizzo email venga immesso tramite un campo di testo apposito.	Non eseguito
TS-RS1O 1.2	Verificare che durante l'autenticazione di un utente, la password venga immessa tramite un capo di testo apposito.	Non eseguito
TS-RS1O 1.3	Verificare che, in caso di fallimento dell'autenticazione di un utente, il sistema visualizzi una pagina di errore.	Non eseguito
TS-RS1O 1.4	Verificare che, in caso di successo dell'autenticazione di un utente, il sistema lo reindirizzi ad una pagina contenente la dashboard attiva.	Non eseguito
TS-RS1O 2.1.1	Verificare che durante la registrazione di un nuovo Proprietario l'indirizzo email venga immesso tramite un campo di testo apposito.	Non eseguito
TS-RS1O 2.1.2	Verificare che durante la registrazione di un nuovo Proprietario la password e la sua conferma vengano immessi tramite dei campi di testo appositi.	Non eseguito
TS-RS1O 2.1.2.1	Verificare che durante la registrazione di un nuovo Proprietario venga richiesto che la password sia composta di un minimo di 6 caratteri.	Non eseguito
TS-RS1O 2.1.2.2	Verificare che durante la registrazione di un nuovo Proprietario venga richiesto che la password sia composta di un massimo di 12 caratteri.	Non eseguito
TS-RS1O 2.1.2.3	Verificare che durante la registrazione di un nuovo Proprietario venga richiesto che la password sia composta da caratteri alfanumerici con possibili caratteri di punteggiatura.	Non eseguito
TS-RS1O 2.1.3	Verificare che durante la registrazione di un nuovo Proprietario il nome dell'azienda da creare venga immesso tramite un campo di testo apposito.	Non eseguito
TS-RS1O 2.2.1	Verificare che durante la registrazione di un nuovo utente invitato la password e la sua conferma vengano immessi tramite dei campi di testo appositi.	Non eseguito



TS-RS1O 2.2.1.1	Verificare che durante la registrazione di un nuovo utente invitato venga richiesto che la password sia composta di un minimo di 6 caratteri.	Non eseguito
TS-RS1O 2.2.1.2	Verificare che durante la registrazione di un nuovo utente invitato venga richiesto che la password sia composta di un massimo di 12 caratteri.	Non eseguito
TS-RS1O 2.2.1.3	Verificare che durante la registrazione di un nuovo utente invitato venga richiesto che la password sia composta da caratteri alfanumerici con possibili caratteri di punteggiatura.	Non eseguito
TS-RS1O 2.3	Verificare che, in caso di fallimento della registrazione di un utente per email non valida, il sistema visualizzi una pagina di errore specificando tale causa.	Non eseguito
TS-RS1O 2.4	Verificare che, in caso di fallimento della registrazione di un utente per password non valida, il sistema visualizzi una pagina di errore specificando se le due password inserite non corrispondono o se la password non soddisfa la struttura richiesta.	Non eseguito
TS-RS1O 2.5	Verificare che, in caso di fallimento della registrazione di un Proprietario per nome azienda già presente, il sistema visualizzi una pagina di errore specificando tale causa.	Non eseguito
TS-RS1O 2.6	Verificare che, in caso di successo della registrazione di un utente, il sistema visualizzi una pagina di avvenuta registrazione.	Non eseguito
TS-RS1D 3.1	Verificare che il sistema permetta di modificare il proprio avatar mediante un apposito bottone che consente di fare l'upload di una nuova immagine.	Non eseguito
TS-RS1O 3.2	Verificare che il sistema permetta di modificare la propria password mediante un apposito campo di testo.	Non eseguito
TS-RS1O 3.3	Verificare che, in caso di fallimento della modifica della password, il sistema visualizzi un messaggio di errore.	Non eseguito
TS-RS1D 3.4	Verificare che il sistema metta a disposizione una pagina per la modifica dei dati anagrafici dell'utente autenticato.	Non eseguito





TS-RS1D 3.4.1	Verificare che, durante la modifica dei dati anagrafici, il sistema metta a disposizione campo di testo per la modifica del nome dell'utente.	Non eseguito
TS-RS1D 3.4.2	Verificare che, durante la modifica dei dati anagrafici, il sistema metta a disposizione campo di testo per la modifica del cognome dell'utente.	Non eseguito
TS-RS1D 3.4.3	Verificare che, durante la modifica dei dati anagrafici, il sistema metta a disposizione campo di testo per la modifica della data di nascita dell'utente.	Non eseguito
TS-RS1D 3.4.4	Verificare che, durante la modifica dei dati anagrafici, il sistema metta a disposizione campo di testo per la modifica del sesso dell'utente.	Non eseguito
TS-RS1O 4.1	Verificare che il sistema metta a disposizione del Proprietario e degli Amministratori la possibilità di connettere un nuovo database e fornisca un campo di testo per l'inserimento del nome del database.	Non eseguito
TS-RS1O 4.2	Verificare che il sistema metta a disposizione del Proprietario e degli Amministratori la possibilità di disconnettere un database e fornisca un bottone apposito.	Non eseguito
TS-RS1O 4.3	Verificare che il sistema metta a disposizione del Proprietario e degli Amministratori la possibilità di modificare i permessi di accesso ad un database e fornisca un'interfaccia apposita, contenente la lista dei membri dell'azienda ed un'apposita checkbox per ognuno di essi.	Non eseguito
TS-RS1O 4.4	Verificare che il sistema visualizzi un messaggio di errore al Proprietario o agli Amministratori che cerchino un database non presente nel sistema.	Non eseguito
TS-RS1O 4.5	Verificare che il sistema permetta al Proprietario o agli Amministratori di cercare un database dell'azienda presente nel sistema.	Non eseguito
TS-RS1F 5.1	Verificare che il sistema fornisca un editor di testo con funzionalità avanzate, come l'e: videnziatura sintattica, l'autocompletamento, l'indentazione ed i suggerimenti.	Non eseguito



TS-RS1O 5.2	Verificare che il sistema permetta di creare nuove DSLIS mediante un'interfaccia apposita contenente l'editor testuale e degli appositi bottoni per compilare e salvare il codice.	Non eseguito
TS-RS1O 5.3	Verificare che il sistema permetta di modificare le DSLIS mediante un'interfaccia apposita contenente l'editor testuale e degli appositi bottoni per compilare e salvare il codice.	Non eseguito
TS-RS1O 5.4	Verificare che il sistema permetta di eliminare una DSLIS mediante un apposito bottone presente in ogni elemento della lista delle istruzioni.	Non eseguito
TS-RS1O 5.5	Verificare che il sistema permetta di visualizzare il codice delle DSLIS mediante un'interfaccia apposita contenente un editor testuale.	Non eseguito
TS-RS1O 5.6	Verificare che il sistema permetta di modificare i permessi di esecuzione, scrittura e lettura dei DSLIS mediante un'apposita interfaccia contenente la lista degli utenti dell'azienda iscritta a MaaS.	Non eseguito
TS-RS1O 5.6.1	Verificare che il sistema permetta di modificare i permessi di esecuzione di un DSLIS mediante un apposito checkbox.	Non eseguito
TS-RS1O 5.6.2	Verificare che il sistema permetta di modificare i permessi di scrittura di un DSLIS mediante un apposito checkbox.	Non eseguito
TS-RS1O 5.6.3	Verificare che il sistema permetta di modificare i permessi di lettura di un DSLIS mediante un apposito checkbox.	Non eseguito
TS-RS1O 5.6.4	Verificare che il sistema fornisca un'interfaccia contenente la lista dei DSLIS a cui l'utente ha accesso.	Non eseguito
TS-RS1O 5.6.5	Verificare che il sistema fornisca una funzionalità di ricerca per sfoltire i dati visualizzati nella lista dei DSLIS.	Non eseguito
TS-RS1O 5.6.6	Verificare che il sistema fornisca un'interfaccia contenente la lista degli utenti dell'azienda iscritta a MaaS.	Non eseguito
TS-RS1O 5.6.7	Verificare che il sistema fornisca una funzionalità di ricerca per sfoltire i dati visualizzati nella lista degli utenti.	Non eseguito



TS-RS1O 5.7	Verificare che il sistema permetta di definire una Dashboard mediante la parola chiave <b>dashboard</b> .	Non eseguito
TS-RS1O 5.7.1	Verificare che il sistema permetta di definire una Row di una Dashboard mediante la parola chiave <b>row</b> .	Non eseguito
TS-RS1O 5.7.1.1	Verificare che il sistema permetta di definire una Collection all'interno di una Row di una Dashboard mediante la parola chiave <b>collection</b> .	Non eseguito
TS-RS1O 5.7.1.2	Verificare che il sistema permetta di definire un Document all'interno di una Row di una Dashboard mediante la parola chiave <b>document</b> .	Non eseguito
TS-RS1O 5.7.1.3	Verificare che il sistema permetta di definire una Cell all'interno di una Row di una Dashboard mediante la parola chiave <b>cell</b> .	Non eseguito
TS-RS1O 5.7.2	Verificare che il sistema permetta di definire il nome di una Dashboard mediante la parola chiave <b>name</b> .	Non eseguito
TS-RS1O 5.8	Verificare che il sistema permetta di definire una Collection mediante la parola chiave <b>collection</b> .	Non eseguito
TS-RS1O 5.8.1	Verificare che il sistema permetta di definire l'identità di una Collection mediante le parole chiave <b>name label</b> , <b>id</b> e <b>weight</b> .	Non eseguito
TS-RS1O 5.8.1.1	Verificare che il sistema permetta di definire il nome di una collection MongoDB, a cui la Collection si deve referenziare, mediante la parola chiave <b>name</b> .	Non eseguito
TS-RS1O 5.8.1.2	Verificare che il sistema permetta di definire il nome da visualizzare di una Collection, mediante la parola chiave <b>label</b> .	Non eseguito
TS-RS1O 5.8.1.3	Verificare che il sistema permetta di definire l'id di una Collection mediante la parola chiave <b>id</b> e che di default esso assuma il valore del campo <b>name</b> .	Non eseguito
TS-RS1O 5.8.1.4	Verificare che il sistema permetta di definire l'ordine di visualizzazione di una Collection all'interno della lista dei DSLIS mediante la parola chiave <b>weight</b> e che di default esso assuma il valore 0.	Non eseguito



TS-RS1O 5.8.2	Verificare che il sistema permetta di definire l'index di una Collection mediante la parola chiave <b>index</b> .	Non eseguito
TS-RS1O 5.8.2.1	Verificare che il sistema permetta di definire gli attributi da visualizzare di una Collection mediante la parola chiave <b>column</b> .	Non eseguito
TS-RS1O 5.8.2.2	Verificare che il sistema permetta di definire l'ordinamento degli attributi di una Collection mediante le parole chiave <b>sortBy</b> e <b>order</b> .	Non eseguito
TS-RS1O 5.8.2.3	Verificare che il sistema permetta di definire il limite di elementi da visualizzare di una Collection mediante la parola chiave <b>perPage</b> .	Non eseguito
TS-RS1O 5.8.2.4	Verificare che il sistema permetta di definire la query di una Collection mediante la parola chiave <b>query</b> .	Non eseguito
TS-RS1O 5.8.2.5	Verificare che il sistema permetta di definire gli attributi selezionabili di una Collection mediante la parola chiave <b>selectable</b> .	Non eseguito
TS-RS1O 5.8.2.6	Verificare che il sistema permetta di definire gli attributi ordinabili di una Collection mediante la parola chiave <b>sortable</b> .	Non eseguito
TS-RS1O 5.8.3	Verificare che il sistema permetta di definire le Action di una Collection mediante la parola chiave <b>action</b> . Verificare inoltre che esso fornisca le parole chiave <b>export</b> e <b>sendEmail</b> per definire le rispettive azioni.	Non eseguito
TS-RS1O 5.9	Verificare che il sistema permetta di definire un Document mediante la parola chiave <b>document</b> .	Non eseguito
TS-RS1O 5.9.1	Verificare che il sistema permetta di definire gli attributi da visualizzare di un Document mediante la parola chiave <b>row</b> .	Non eseguito
TS-RS1O 5.9.2	Verificare che il sistema permetta di definire il tipo degli attributi di un Document mediante la parola chiave <b>type</b> e fornisca inoltre le parole chiave <b>array</b> , <b>object</b> , <b>link</b> e <b>image</b> per definire i vari tipi.	Non eseguito
TS-RS1O 5.9.3	Verificare che il sistema permetta di definire le Action di un Document mediante la parola chiave <b>action</b> . Verificare inoltre che esso fornisca le parole chiave <b>export</b> e <b>sendEmail</b> per definire le rispettive azioni.	Non eseguito



TS-RS1O 5.10	Verificare che il sistema permetta di definire una Cell mediante la parola chiave <b>cell</b> .	Non eseguito
TS-RS1O 5.10.1	Verificare che il sistema permetta di definire la query di una Cell mediante la parola chiave <b>query</b> .	Non eseguito
TS-RS1O 5.10.2	Verificare che il sistema permetta di definire l'ordinamento dei risultati della query di una Cell mediante le parole chiave <b>sortBy</b> e <b>order</b> .	Non eseguito
TS-RS1O 5.10.3	Verificare che il sistema permetta di definire il tipo di una Cell mediante la parola chiave <b>type</b> e fornisca inoltre le parole chiave <b>string</b> , <b>number</b> , <b>link</b> , <b>image</b> e <b>date</b> per definire i vari tipi.	Non eseguito
TS-RS1O 5.10.4	Verificare che il sistema permetta di definire il valore arbitrario di una Cell mediante la parola chiave <b>value</b> .	Non eseguito
TS-RS1O 5.11	Verificare che il sistema permetta di definire una Action mediante la parola chiave <b>action</b> .	Non eseguito
TS-RS1O 5.11.1	Verificare che il sistema permetta di definire l'Export di una Action mediante la parola chiave <b>export</b> .	Non eseguito
TS-RS1O 5.11.2	Verificare che il sistema permetta di definire la Send Email di una Action mediante la parola chiave <b>sendEmail</b> .	Non eseguito
TS-RS1O 5.12	Verificare che il sistema visualizzi un messaggio di errore nel caso in cui l'utente cerchi un DSLIS non memorizzato al suo interno.	Non eseguito
TS-RS1O 5.13	Verificare che il sistema visualizzi un messaggio di errore nel caso in cui l'utente compili un DSLIS in cui sono presenti errori di sintassi.	Non eseguito
TS-RS1O 5.14	Verificare che il sistema permetta di eseguire un DSLIS, mettendo a disposizione dell'utente un'interfaccia apposita contenente la lista dei DSLIS a cui ha accesso.	Non eseguito
TS-RS1O 5.14.1	Verificare che il sistema consenta all'utente di eseguire un'Action sui dati visualizzati da un DSLIS, fornendo degli appositi bottoni.	Non eseguito
TS-RS1O 5.14.1.1	Verificare che il sistema consenta all'utente di eseguire un'Export sui dati visualizzati da un DSLIS, fornendo degli appositi bottoni.	Non eseguito



TS-RS1O 5.14.1.2	Verificare che il sistema consenta all'utente di eseguire una Send Email sui dati visualizzati da un DSLIS, fornendo degli appositi bottoni.	Non eseguito
TS-RS1O 5.15	Verificare che il sistema permetta di importare un DSLIS, fornendo un bottone apposito.	Non eseguito
TS-RS1O 5.16	Verificare che il sistema permetta di esportare un DSLIS, fornendo un bottone apposito.	Non eseguito
TS-RS1O 5.17	Verificare che il sistema permetta di ricercare un DSLIS, tra quelli a cui l'utente ha accesso.	Non eseguito
TS-RS1O 5.18	Verificare che il sistema permetta di visualizzare la lista completa dei DSLIS a cui l'utente ha accesso.	Non eseguito
TS-RS1O 6.1	Verificare che il sistema permetta al Proprietario e agli Amministratori di rimuovere un utente dell'azienda, mediante un'apposita interfaccia grafica.	Non eseguito
TS-RS1O 6.1.1	Verificare che il sistema permetta al Proprietario e agli Amministratori di rimuovere un Amministratore dell'azienda.	Non eseguito
TS-RS1O 6.1.2	Verificare che il sistema permetta al Proprietario e agli Amministratori di rimuovere un Membro dell'azienda.	Non eseguito
TS-RS1O 6.1.3	Verificare che il sistema permetta al Proprietario e agli Amministratori di rimuovere un Ospite dell'azienda.	Non eseguito
TS-RS1O 6.2	Verificare che il sistema permetta al Proprietario e agli Amministratori di aggiungere un utente all'azienda, mediante un'apposita interfaccia grafica.	Non eseguito
TS-RS1O 6.2.1	Verificare che il sistema permetta al Proprietario e agli Amministratori di invitare un nuovo Amministratore all'azienda, mediante un campo di testo per l'inserimento dell'email.	Non eseguito
TS-RS1O 6.2.2	Verificare che il sistema permetta al Proprietario e agli Amministratori di invitare un nuovo Membro all'azienda, mediante un campo di testo per l'inserimento dell'email.	Non eseguito
TS-RS1O 6.2.3	Verificare che il sistema permetta al Proprietario e agli Amministratori di invitare un nuovo Ospite all'azienda, mediante un campo di testo per l'inserimento dell'email.	Non eseguito



TS-RS1O 6.3	Verificare che il sistema permetta al Proprietario e agli Amministratori di modificare il ruolo di un utente dell'azienda.	Non eseguito
TS-RS1O 6.3.1	Verificare che il sistema permetta al Proprietario e agli Amministratori di promuovere un utente dell'azienda ad un ruolo superiore.	Non eseguito
TS-RS1O 6.3.1.1	Verificare che il sistema permetta al Proprietario e agli Amministratori di promuovere un Ospite o un Membro dell'azienda al ruolo di Amministratore.	Non eseguito
TS-RS1O 6.3.1.2	Verificare che il sistema permetta al Proprietario e agli Amministratori di promuovere un Ospite dell'azienda al ruolo di Membro.	Non eseguito
TS-RS1O 6.3.2	Verificare che il sistema permetta al Proprietario e agli Amministratori di retrocedere un utente dell'azienda ad un ruolo inferiore.	Non eseguito
TS-RS1O 6.3.2.1	Verificare che il sistema permetta al Proprietario e agli Amministratori di retrocedere un qualsiasi utente dell'azienda (escluso il Proprietario) al ruolo di Ospite.	Non eseguito
TS-RS1O 6.3.2.2	Verificare che il sistema permetta al Proprietario e agli Amministratori di retrocedere un qualsiasi utente dell'azienda (escluso il Proprietario) al ruolo di Membro.	Non eseguito
TS-RS1O 7	Verificare che il sistema permetta all'utente autenticato di effettuare il logout mediante un apposito bottone presente in ogni interfaccia di MaaS.	Non eseguito
TS-RS1O 8.1	Verificare che il sistema permetta all'utente non autenticato di recuperare la propria password, mediante un'apposita interfaccia contenente un campo di testo per l'inserimento dell'email.	Non eseguito
TS-RS1O 8.2	Verificare che il sistema, durante il recupero della password, invii un'email contenente un link attraverso il quale l'utente non autenticato possa effettuare il reset della propria password.	Non eseguito
TS-RS1O 8.3	Verificare che il sistema permetta ad un utente non autenticato che abbia richiesto il recupero della password, di accedere ad un'apposita interfaccia che consenta il reset della propria password, tramite il link ricevuto nell'email di recupero.	Non eseguito



TS-RS1O 8.3.1	Verificare che il sistema permetta ad un utente non autenticato che abbia richiesto il recupero della password, di scegliere una nuova password che sostituirà la precedente e di confermarla, mediante dei campi di testo appositi.	Non eseguito
TS-RS1O 8.3.1.1	Verificare che durante il recupero password di un utente venga richiesto che la password sia composta di un minimo di 6 caratteri.	Non eseguito
TS-RS1O 8.3.1.2	Verificare che durante il recupero password di un utente venga richiesto che la password sia composta di un massimo di 12 caratteri.	Non eseguito
TS-RS1O 8.3.1.3	Verificare che durante il recupero password di un utente venga richiesto che la password sia composta da caratteri alfanumerici con possibili caratteri di punteggiatura.	Non eseguito
TS-RS1O 8.4	Verificare che durante il recupero password di un utente il sistema visualizzi un messaggio di errore se l'email inserita per il recupero non rientra tra quelle degli iscritti a MaaS.	Non eseguito
TS-RS1O 8.5	Verificare che durante il recupero password di un utente il sistema visualizzi un messaggio di errore se la password inserita per il recupero non rispetta la struttura richiesta oppure se non corrisponde a quella inserita per la conferma.	Non eseguito
TS-RS1O 8.6	Verificare che il sistema visualizzi un messaggio di avvenuto reset della password nel caso in cui il suo recupero vada a buon fine.	Non eseguito
TS-RS1O 9.1	Verificare che il sistema permetta all'utente di modificare la Dashboard attiva mediante un'apposita interfaccia contenente la lista di quelle a cui ha accesso.	Non eseguito
TS-RS1O 9.2	Verificare che durante la modifica della Dashboard attiva il sistema permetta all'utente di cercare le Dashboard tra quelle a cui ha accesso.	Non eseguito
TS-RS1O 9.3	Verificare che durante la modifica della Dashboard attiva di un utente il sistema visualizzi un messaggio di errore se la Dashboard cercata non rientra tra quelle a cui ha accesso.	Non eseguito
TS-RS1O 10	Verificare che il sistema permetta all'utente autenticato di visualizzare la propria Dashboard attiva mediante un'apposita interfaccia.	Non eseguito





TS-RS1O 11	Verificare che il sistema visualizzi all'utente un messaggio di errore per aver cercato un utente non presente nel sistema MaaS.	Non eseguito
TS-RS1O 12	Verificare che il sistema permetta di cercare un utente attraverso un'apposita interfaccia, contenente un campo di testo in cui inserire il nome o l'email dell'utente da cercare.	Non eseguito
TS-RS1O 13.1	Verificare che il sistema permetta al Super-Amministratore di gestire le aziende registrate, mediante un'apposita interfaccia grafica.	Non eseguito
TS-RS1O 13.1.1	Verificare che il sistema consenta al Super-Amministratore di visualizzare la lista delle aziende registrate.	Non eseguito
TS-RS1O 13.1.2	Verificare che il sistema consenta al Super-Amministratore di ricercare un'azienda tra quelle registrate.	Non eseguito
TS-RS1O 13.1.3	Verificare che il sistema visualizzi al Super-Amministratore un messaggio di errore per aver cercato un'azienda non presente tra quelle registrate.	Non eseguito
TS-RS1O 13.1.4	Verificare che il sistema consenta al Super-Amministratore di modificare il nome di un'azienda mediante un apposito campo di testo.	Non eseguito
S-RS1O 13.1.5	Verificare che il sistema consenta al Super-Amministratore di eliminare un'azienda mediante un apposito bottone.	Non eseguito
TS-RS1O 13.2	Verificare che il sistema permetta al Super-Amministratore di gestire gli utenti registrati, mediante un'apposita interfaccia grafica.	Non eseguito
TS-RS1O 13.2.1	Verificare che il sistema consenta al Super-Amministratore di visualizzare la lista degli utenti registrati.	Non eseguito
TS-RS1O 13.2.2	Verificare che il sistema visualizzi al Super-Amministratore un messaggio di errore per aver cercato un utente non presente tra quelli registrati.	Non eseguito
TS-RS1O 13.2.3	Verificare che il sistema consenta al Super-Amministratore di modificare l'email di un utente mediante un apposito campo di testo.	Non eseguito



TS-RS1O 13.2.4	Verificare che il sistema consenta al Super-Amministratore di modificare la password di un utente mediante un apposito campo di testo.	Non eseguito
TS-RS1O 13.2.5	Verificare che il sistema consenta al Super-Amministratore di eliminare un utente mediante un apposito bottone.	Non eseguito
TS-RS1O 14.1	Verificare che il sistema permetta al Super-Amministratore di impersonificare un utente registrato, mediante un'apposita interfaccia grafica che permetta di selezionarlo.	Non eseguito
TS-RS1O 14.1.1	Verificare che il sistema permetta al Super-Amministratore di impersonificare un Proprietario.	Non eseguito
TS-RS1O 14.1.2	Verificare che il sistema permetta al Super-Amministratore di impersonificare un Amministratore.	Non eseguito
TS-RS1O 14.1.3	Verificare che il sistema permetta al Super-Amministratore di impersonificare un Membro.	Non eseguito
TS-RS1O 14.1.4	Verificare che il sistema permetta al Super-Amministratore di impersonificare un Ospite.	Non eseguito
TS-RS1O 14.2	Verificare che il sistema permetta al Super-Amministratore di visualizzare la lista degli utenti registrati.	Non eseguito
TS-RS1O 14.3	Verificare che il sistema permetta al Super-Amministratore di ricercare un utente tra quelli registrati, mediante un'apposita barra di ricerca.	Non eseguito
TS-RS1O 15.1	Verificare che durante l'autenticazione di un Super-Amministratore l'indirizzo email venga immesso tramite un campo di testo apposito.	Non eseguito
TS-RS1O 15.2	Verificare che durante l'autenticazione di un Super-Amministratore la password venga immessa tramite un campo di testo apposito.	Non eseguito
TS-RS1O 15.3	Verificare che, in caso di fallimento dell'autenticazione di un Super-Amministratore, il sistema visualizzi una pagina di errore.	Non eseguito
TS-RS1O 16	Verificare che il sistema permetta al Super-Amministratore di eseguire il logout mediante un apposito bottone presente in tutte le interfacce.	Non eseguito



TS-RS1O 17	Verificare che il sistema visualizzi al Super-Am:ministratore un messaggio di errore nel caso in cui abbia ricercato un utente non presente nel sistema.	Non eseguito
TS-RS1O 18	Verificare che il sistema permetta al Super-Am:ministratore di ricercare un utente mediante l'inserimento del nome o dell'email in una barra di ricerca apposita.	Non eseguito
TS-RS2O 1	Verificare che il sistema suddivida i risultati derivanti da grandi quantità di dati elaborati in diverse pagine, per garantire la sua stabilità.	Non eseguito
TS-RS4O 1	Verificare che il sistema venga implementato nella piattaforma Heroku.	Non eseguito
TS-RS4O 2	Verificare che il sistema venga sviluppato su un framework Node.js di alto livello, ad esempio LoopBack.	Non eseguito
TS-RS4O 3	Verificare che il sistema venga pubblicato e versionato in una repository di GitHub o BitBucket.	Non eseguito
TS-RS4O 4	Verificare che il sistema si basi su un database sviluppato in MongoDB di versione 3.x o superiore.	Non eseguito
TS-RS4O 5	Verificare che il sistema utilizzi Node.js come back-end in versione LTS (Long Term Support) Argon (v4.2.0).	Non eseguito
TS-RS4O 6	Verificare tutte le password del sistema MaaS siano persistenti e criptate in qualsiasi momento, usando la libreria scrypt.	Non eseguito
TS-RS4O 7	Verificare che il sistema si avvalga dei JWT (JSON Web Tokens) per i link del ripristino della password e per l'invito di nuovi utenti.	Non eseguito
TS-RS4O 8	Verificare che il sistema verifichi l'autenticità di un utente tramite il controllo di email e password confrontando i dati con quelli presenti nel database interno di MaaS.	Non eseguito
TS-RS4O 9	Verificare che il sistema verifichi la correttezza de dati inseriti dall'utente in fase di registrazione.	Non eseguito

Tabella A.2: Test di Sistema

## A.4 Test di integrazione

I test di integrazione servono per verificare che tutti i diversi componenti del sistema comunichino correttamente tra di loro, e che vi sia all'interno del software il flusso di dati atteso. Verrà utilizzata una strategia di integrazione incrementale per poter sviluppare e verificare più componenti in parallelo. Questo metodo permette di dare priorità ai test relativi alle componenti che vengono ritenute più importanti. In questo modo sarà possibile partire dalle componenti che soddisfano i requisiti obbligatori fino ad integrarli con le componenti che soddisfano i requisiti opzionali. Permette anche di restringere la ricerca dell'errore in caso di test fallito, in quanto sarà molto probabile che l'errore risulti dal nuovo componente o dalle sue interazioni con il sistema corrente. Non sarà da escludere il caso in cui il test fallisca per il fatto che la nuova istanza di test adoperi un campione di input non precedentemente trattato, portando così il sistema a generare un errore.

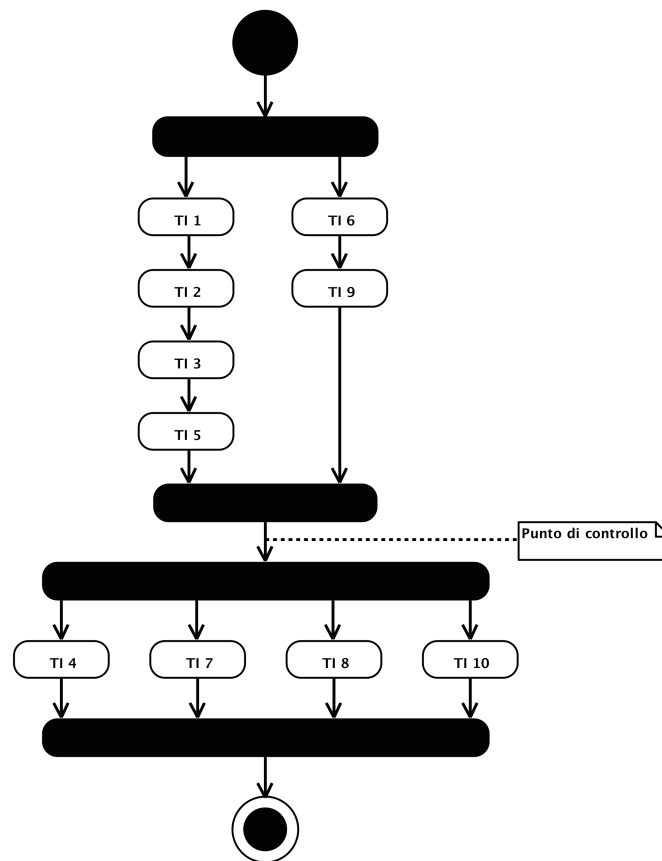


Figura 3: Sequenza d'integrazione delle componenti

Si è scelto di utilizzare un approccio top-down il quale garantisce che le componenti di più alto livello siano testate non appena sono implementate. Man mano che si procede con la codifica delle componenti di più basso livello, queste vengono integrate e viene eseguito il relativo test.



Test	Descrizione	Componenti aggiunte	Stato
TI 1	Si verifica che l'applicazione Web carichi correttamente le librerie JavaScript utilizzate.	Front-end	Non eseguito.
TI 2	Si verifica che sia funzionante il collegamento con il server e con l'ActionCreators.	Front-end::WebAPIUtils	Non eseguito.
TI 3	Si verifica che l'ActionCreators riceva correttamente gli input dalla View e crei le action ad essi associate.	Front-end::ActionCreators	Non eseguito.
TI 4	Si verifica che gli Stores sia caricati correttamente e comunichino i cambiamenti alla View.	Front-end::Stores	Non eseguito.
TI 5	Si verifica che la View funzioni correttamente permettendo il caricamento e la visualizzazione di tutte le pagine.	Front-End::View	Non eseguito.
TI 6	Si verifica che il Middleware si integri correttamente nella gestione delle richieste che arrivano al server.	Back-end::Middlewares	Non eseguito.
TI 7	Si verifica che il Model si integri correttamente nella gestione dell'inserimento, della modifica, della creazione e dell'eliminazione consistente dei dati.	Back-end::Model	Non Eseguito.
TI 8	Si verifica che il Datasource si integri correttamente nella gestione dei Database esterni.	Back-end::Datasource	Non eseguito
TI 9	Si verifica che le RESTAPIs mappino correttamente le richieste HTTP da parte del client.	Back-end::RESTAPIs	Non eseguito.
TI 10	Si verifica che il Connectors si connetta correttamente ai database esterni e al servizio di posta elettronica.	Back-end::Connectors	Non eseguito

Tabella A.3: Descrizione test d'Integrazione



## A.5 Test di unità

Di seguito vengono elencati i test di unità pianificati. Si ritiene sia strettamente necessario, per comprendere interamente il contenuto della seguente tabella, leggere queste informazioni avendo a portata di mano il *tracciamento metodi-test*, il quale può essere trovato all'interno del documento *Definizione di Prodotto v1.0.0*.

Nome	Descrizione	Stato
TU - 1	Verifica che un utente sia stato inserito correttamente nel database di MaaS e che non ci siano errore.	Non eseguito.
TU - 2	Verifica che un utente si a riuscito ad autenticarsi al sistema MaaS, senza che siano avvenuti errori che non dipendono da lui.	Non eseguito.
TU - 3	Verifica della corretta esecuzione di una modifica su un utente.	Non eseguito.
TU - 4	Verifica che un utente riesca a disconnettersi senza che avvengano errori.	Non eseguito.
TU - 5	Verifica che si riescano ad ottenere i dati degli utenti di un'azienda.	Non eseguito.
TU - 6	Verifica che si riescano ad ottenere i dati relativi ad un utente di un'azienda.	Non eseguito.
TU - 7	Controllo dell'effettiva eliminazione di un utente dal database di MaaS.	Non eseguito.
TU - 8	Verifica che si riesca ad ottenere una lista di utenti aventi un determinato valore tra i propri dati.	Non eseguito.
TU - 9	Controllo che un utente che ha dimenticato la propria password riesca ad accedere al sistema di recupero password.	Non eseguito.
TU - 10	Verifica che si riesca a cambiare il ruolo di un utente.	Non eseguito.
TU - 11	Verifica che si riesca a generare (tramite l'Action <code>sendEmail</code> ) una mail di invito per un nuovo utente e ad inviarla.	Non eseguito.
TU - 12	Controllo della possibilità di effettuare il login da parte del Super Amministratore.	Non eseguito.
TU - 13	Controllo che il Super Amministratore riesca a disconnettersi dal sistema, senza che avvengano errori.	Non eseguito.
TU - 14	Si verifica che il Super Amministratore possa impersonare un utente iscritto al sistema, assumendone quindi identità, ruolo e privilegi.	Non eseguito.



TU - 15	Si controlla che un Super Amministratore possa, tramite questo metodo, visualizzare una lista contenente tutte le aziende iscritte al sistema.	Non eseguito.
TU - 16	Si controlla che un possibile Proprietario possa creare un'azienda ed inserirla nel database di MaaS.	Non eseguito.
TU - 17	Viene verificato il funzionamento del metodo che restituisce al Super Amministratore le informazioni relative ad una specifica azienda.	Non eseguito.
TU - 18	Viene verificato il funzionamento di un metodo che permette al Super Amministratore di modificare i dati di un'azienda, a scopo di supporto, qualora un Proprietario non riesca a farlo.	Non eseguito.
TU - 19	Si controlla che un Super Amministratore riesca ad eliminare un'azienda dal database del sistema MaaS.	Non eseguito.
TU - 20	Verifica della possibilità, da parte di un Super Amministratore, di ottenere le informazioni relative ad un'azienda.	Non eseguito.
TU - 21	Si verifica che un Amministratore o Proprietario riesca a collegare un database esterno alla propria azienda, senza che si verifichino errori durante il processo.	Non eseguito.
TU - 22	Si controlla che un Amministratore o Proprietario possa eliminare l'origine dei dati (database esterno) di un'azienda, senza che si verifichino errori durante il processo.	Non eseguito.
TU - 23	Verifica che gli Amministratori di un'azienda iscritta al sistema MaaS siano in grado ottenere specifici dati, appartenenti alla propria azienda, conservati in un database esterno collegato.	Non eseguito.
TU - 24	Si controlla che gli Amministratori di un'azienda possano cercare un database esterno collegato.	Non eseguito.
TU - 25	Si controlla che un Amministratore di un'azienda possa concedere i permessi di accesso al database esterno ad un utente iscritto alla stessa azienda.	Non eseguito.
TU - 26	Si controlla che gli Amministratori di un'azienda possa negare i permessi di accesso al database esterno ad un utente iscritto alla stessa azienda.	Non eseguito.
TU - 27	Si verifica che un utente possa visualizzare una lista contenente tutte le Collection a cui ha accesso.	Non eseguito.
TU - 28	Si verifica che un utente possa creare una nuova Collection, senza che si presentino errori.	Non eseguito.



TU - 29	Si controlla che un utente possa modificare una Collection a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 30	Si controlla che un utente possa eliminare una Collection a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 31	Si controlla che un qualsivoglia utente possa ottenere il codice di una Collection, ossia il DSLIS da cui essa ha avuto origine.	Non eseguito.
TU - 32	Si verifica la possibilità da parte di un utente di ricercare un insieme di Collection, filtrando i risultati in base ad un valore.	Non eseguito.
TU - 33	Si controlla che un utente possa eseguire una Collection (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori.	Non eseguito.
TU - 34	Verifica che il metodo abbia inviato via mail, tramite una Action <code>sendEmail</code> , le informazioni renderizzate da una Collection.	Non eseguito.
TU - 35	Si verifica che un utente possa esportare i dati visualizzati in una Collection, senza che si verifichino errori.	Non eseguito.
TU - 36	Si controlla che un utente possa esportare il codice DSLIS generatore di una Collection, senza che si verifichino errori.	Non eseguito.
TU - 37	Si controlla che un utente possa importare il codice DSLIS generatore di una Collection.	Non eseguito.
TU - 38	Si verifica che un utente possa visualizzare una lista contenente tutti i Document a cui ha accesso.	Non eseguito.
TU - 39	Si verifica che un utente possa creare un nuovo Document, senza che si presentino errori.	Non eseguito.
TU - 40	Si controlla che un utente possa modificare un Document a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 41	Si controlla che un utente possa eliminare un Document a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 42	Si verifica la possibilità da parte di un utente di ricercare un insieme di Document, filtrando i risultati in base ad un valore.	Non eseguito.
TU - 43	Si controlla che un utente possa eseguire un Document (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori.	Non eseguito.





TU - 44	Si controlla che un utente possa eseguire un Document appartenente ad una Collection (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori.	Non eseguito.
TU - 45	Verifica che il metodo abbia inviato via mail, tramite una Action <code>sendEmail</code> , le informazioni renderizzate da un Document.	Non eseguito.
TU - 46	Si verifica che un utente possa esportare i dati visualizzati in un Document, senza che si verifichino errori.	Non eseguito.
TU - 47	Si controlla che un qualsivoglia utente possa ottenere il codice di un Document, ossia il DSLIS da cui esso ha avuto origine.	Non eseguito.
TU - 48	Si controlla che un utente possa esportare il codice DSLIS generatore di un Document, senza che si verifichino errori.	Non eseguito.
TU - 49	Si controlla che un utente possa importare il codice DSLIS generatore di un Document.	Non eseguito.
TU - 50	Si verifica che un utente possa visualizzare una lista contenente tutte le Dashboard a cui ha accesso.	Non eseguito.
TU - 51	Si verifica che un utente possa creare una nuova Dashboard, senza che si presentino errori.	Non eseguito.
TU - 52	Si controlla che un utente possa modificare una Dashboard a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 53	Si controlla che un utente possa eliminare una Dashboard a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 54	Si controlla che un qualsivoglia utente possa ottenere il codice di una Dashboard, ossia il DSLIS da cui essa ha avuto origine.	Non eseguito.
TU - 55	Si verifica la possibilità da parte di un utente di ricercare un insieme di Dashboard, filtrando i risultati in base ad un valore.	Non eseguito.
TU - 56	Si controlla che un utente possa eseguire una Dashboard (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori.	Non eseguito.
TU - 57	Si controlla che un utente possa esportare il codice DSLIS generatore di una Dashboard, senza che si verifichino errori.	Non eseguito.
TU - 58	Si controlla che un utente possa importare il codice DSLIS generatore di una Dashboard.	Non eseguito.



TU - 59	Si verifica che un utente possa visualizzare una lista contenente tutte le Cell a cui ha accesso.	Non eseguito.
TU - 60	Si verifica che un utente possa creare una nuova Cell, senza che si presentino errori.	Non eseguito.
TU - 61	Si controlla che un utente possa modificare una Cell a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 62	Si controlla che un utente possa eliminare una Cell a cui ha accesso, qualora avesse permessi sufficienti.	Non eseguito.
TU - 63	Si verifica la possibilità da parte di un utente di ricercare un insieme di Cell, filtrando i risultati in base ad un valore.	Non eseguito.
TU - 64	Si controlla che un utente possa eseguire una Cell (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori.	Non eseguito.
TU - 65	Si controlla che un qualsivoglia utente possa ottenere il codice di una Cell, ossia il DSLIS da cui essa ha avuto origine.	Non eseguito.
TU - 66	Si controlla che un utente possa esportare il codice DSLIS generatore di una Cell, senza che si verifichino errori.	Non eseguito.
TU - 67	Si controlla che un utente possa importare il codice DSLIS generatore di una Cell.	Non eseguito.
TU - 68	Verifica che, all'atto dell'esportazione del codice di un DSLIS corretto, venga effettuata una conversione ugualmente corretta di esso nel formato JSON.	Non eseguito.
TU - 69	Verifica che, all'atto dell'esportazione del codice di un DSLIS corretto, venga effettuata una conversione ugualmente corretta di esso nel formato CSV.	Non eseguito.
TU - 70	Si verifica che funzioni correttamente la comunicazione tra front-end e back-end tramite i middleware.	Non eseguito.
TU - 71	Si verifica che funzioni correttamente il controllo delle richieste fatte dal client al server, contenenti url errati.	Non eseguito.
TU - 72	Si verifica che funzioni correttamente il controllo delle richieste fatte dal client al server, nel momento in cui quest'ultimo non sia in grado di gestirle.	Non eseguito.
TU - 73	Si controlla che il server venga inizializzato correttamente.	Non eseguito.



TU - 74	Verifica che sia possibile inviare tramite HTTP una richiesta per la sostituzione dell'avatar di un utente, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 75	Verifica che sia possibile inviare tramite HTTP una richiesta per la modifica dei dati personali di un utente, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 76	Verifica che sia possibile inviare tramite HTTP una richiesta per la sostituzione della password di un utente, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 77	Verifica che sia possibile inviare tramite HTTP una richiesta per la visualizzazione degli utenti di un'azienda, oltre che dei loro dati.	Non eseguito.
TU - 78	Verifica che sia possibile inviare tramite HTTP una richiesta per la visualizzazione dei dati di un utente.	Non eseguito.
TU - 79	Verifica che sia possibile inviare tramite HTTP una richiesta per l'eliminazione di un utente, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 80	Verifica che sia possibile inviare tramite HTTP una richiesta per la ricerca di un insieme di utenti, aventi una determinata proprietà, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 81	Verifica che sia possibile inviare tramite HTTP una richiesta per per il sistema di recupero password, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 82	Verifica che sia possibile inviare tramite HTTP una richiesta per la modifica del ruolo di un utente, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 83	Verifica che sia possibile inviare tramite HTTP una richiesta per l'invito di un potenziale utente del sistema MaaS (con relativo ruolo), e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 84	Si verifica che, attraverso una richiesta HTTP, si possa visualizzare una lista contenente tutte le Collection a cui si ha accesso, e che sia possibile ricevere dal server un messaggio d'errore.	Non eseguito.



TU - 85	Si verifica che, attraverso una richiesta HTTP, si possa creare una nuova Collection, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 86	Si controlla che, attraverso una richiesta HTTP, si possa modificare una Collection a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 87	Si controlla che, attraverso una richiesta HTTP, si possa eliminare una Collection a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 88	Si controlla che, attraverso una richiesta HTTP, si possa ottenere il codice di una Collection, ossia il DSLIS da cui essa ha avuto origine, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 89	Si verifica la possibilità, attraverso una richiesta HTTP, di ricercare un insieme di Collection, filtrando i risultati in base ad un valore, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 90	Si controlla che, attraverso una richiesta HTTP, si possa eseguire una Collection (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori, e che sia possibile ricevere dal server il risultato dell'operazione in formato JSON.	Non eseguito.
TU - 91	Verifica che, attraverso una richiesta HTTP, il metodo abbia inviato via mail, tramite una Action <code>sendEmail</code> , le informazioni renderizzate da una Collection, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 92	Si verifica che, attraverso una richiesta HTTP, si possano esportare i dati visualizzati in una Collection, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 93	Si controlla che, attraverso una richiesta HTTP, si possa esportare il codice DSLIS generatore di una Collection, e che sia possibile ricevere dal server un messaggio di buona riuscita.	Non eseguito.
TU - 94	Si controlla che, attraverso una richiesta HTTP, si possa importare il codice DSLIS generatore di una Collection, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.



TU - 95	Si verifica che, attraverso una richiesta HTTP, si possa visualizzare una lista contenente tutti i Document a cui si ha accesso, e che sia possibile ricevere dal server un messaggio d'errore.	Non eseguito.
TU - 96	Si verifica che, attraverso una richiesta HTTP, si possa creare un nuovo Document, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 97	Si controlla che, attraverso una richiesta HTTP, si possa modificare un Document a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 98	Si controlla che, attraverso una richiesta HTTP, si possa eliminare un Document a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 99	Si controlla che, attraverso una richiesta HTTP, si possa ottenere il codice di un Document, ossia il DSLIS da cui essa ha avuto origine, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 100	Si verifica la possibilità, attraverso una richiesta HTTP, di ricercare un insieme di Document, filtrando i risultati in base ad un valore, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 101	Si controlla che, attraverso una richiesta HTTP, si possa eseguire un Document (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori, e che sia possibile ricevere dal server il risultato dell'operazione in formato JSON.	Non eseguito.
TU - 102	Verifica che, attraverso una richiesta HTTP, il metodo abbia inviato via mail, tramite una Action <code>sendEmail</code> , le informazioni renderizzate da una Document, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 103	Si verifica che, attraverso una richiesta HTTP, si possano esportare i dati visualizzati in un Document, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 104	Si controlla che, attraverso una richiesta HTTP, si possa esportare il codice DSLIS generatore di un Document, e che sia possibile ricevere dal server un messaggio di buona riuscita.	Non eseguito.



TU - 105	Si controlla che, attraverso una richiesta HTTP, si possa importare il codice DSLIS generatore di un Document, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 106	Si verifica che, attraverso una richiesta HTTP, si possa visualizzare una lista contenente tutte le Dashboard a cui si ha accesso, e che sia possibile ricevere dal server un messaggio d'errore.	Non eseguito.
TU - 107	Si verifica che, attraverso una richiesta HTTP, si possa creare una nuova Dashboard, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 108	Si controlla che, attraverso una richiesta HTTP, si possa modificare una Dashboard a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 109	Si controlla che, attraverso una richiesta HTTP, si possa eliminare una Dashboard a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 110	Si verifica la possibilità, attraverso una richiesta HTTP, di ricercare un insieme di Dashboard, filtrando i risultati in base ad un valore, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 111	Si controlla che, attraverso una richiesta HTTP, si possa eseguire una Dashboard (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori, e che sia possibile ricevere dal server il risultato dell'operazione in formato JSON.	Non eseguito.
TU - 112	Si controlla che, attraverso una richiesta HTTP, si possa esportare il codice DSLIS generatore di una Dashboard, e che sia possibile ricevere dal server un messaggio di buona riuscita.	Non eseguito.
TU - 113	Si controlla che, attraverso una richiesta HTTP, si possa ottenere il codice di una Dashboard, ossia il DSLIS da cui essa ha avuto origine, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 114	Si controlla che, attraverso una richiesta HTTP, si possa importare il codice DSLIS generatore di una Dashboard, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.



TU - 115	Si verifica che, attraverso una richiesta HTTP, si possa visualizzare una lista contenente tutte le Cell a cui si ha accesso, e che sia possibile ricevere dal server un messaggio d'errore.	Non eseguito.
TU - 116	Si verifica che, attraverso una richiesta HTTP, si possa creare una nuova Cell, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 117	Si controlla che, attraverso una richiesta HTTP, si possa modificare una Cell a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 118	Si controlla che, attraverso una richiesta HTTP, si possa eliminare una Cell a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 119	Si verifica la possibilità, attraverso una richiesta HTTP, di ricercare un insieme di Cell, filtrando i risultati in base ad un valore, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 120	Si controlla che, attraverso una richiesta HTTP, si possa eseguire una Cell (ovvero il suo DSLIS) a cui ha accesso, qualora avesse permessi sufficienti, senza che avvengano errori, e che sia possibile ricevere dal server il risultato dell'operazione in formato JSON.	Non eseguito.
TU - 121	Si controlla che, attraverso una richiesta HTTP, si possa esportare il codice DSLIS generatore di una Cell, e che sia possibile ricevere dal server un messaggio di buona riuscita.	Non eseguito.
TU - 122	Si controlla che, attraverso una richiesta HTTP, si possa ottenere il codice di una Cell, ossia il DSLIS da cui essa ha avuto origine, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 123	Si controlla che, attraverso una richiesta HTTP, si possa importare il codice DSLIS generatore di una Cell, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 124	Si verifica che, attraverso una richiesta HTTP, si possa creare un nuovo database esterno, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.



TU - 125	Si controlla che, attraverso una richiesta HTTP, si possa eliminare un database esterno a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 126	Si verifica che, attraverso una richiesta HTTP, si possa visualizzare una lista contenente tutti i database esterni a cui si ha accesso, e che sia possibile ricevere dal server un messaggio d'errore.	Non eseguito.
TU - 127	Si verifica la possibilità, attraverso una richiesta HTTP, di ricercare un insieme di database esterni, filtrando i risultati in base ad un valore, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 128	Si verifica che, attraverso una richiesta HTTP, si possano attribuire ad un utente permessi di accesso ad un database esterno, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 129	Si verifica che, attraverso una richiesta HTTP, si possano negare ad un utente i permessi di accesso ad un database esterno, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 130	Si verifica che, attraverso una richiesta HTTP, si possa visualizzare una lista contenente tutte le Company a cui si ha accesso, e che sia possibile ricevere dal server un messaggio d'errore.	Non eseguito.
TU - 131	Si verifica che, attraverso una richiesta HTTP, si possa creare una nuova Company, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 132	Si verifica che, attraverso una richiesta HTTP, si possano visualizzare i dati di una determinata Company a cui si ha accesso, e che sia possibile ricevere dal server il risultato dell'operazione.	Non eseguito.
TU - 133	Si controlla che, attraverso una richiesta HTTP, si possa modificare una Company a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 134	Si controlla che, attraverso una richiesta HTTP, si possa eliminare una Company a cui si ha accesso, qualora si avessero permessi sufficienti, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.





TU - 135	Si verifica la possibilità, attraverso una richiesta HTTP, di ricercare un insieme di Company, filtrando i risultati in base ad un valore, e che sia possibile ricevere dal server una lista o un messaggio d'errore.	Non eseguito.
TU - 136	Si controlla che, attraverso una richiesta HTTP, il Super Amministratore riesca ad impersonare un qualsiasi utente del sistema MaaS, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 137	Si controlla che, attraverso una richiesta HTTP, un utente possa essere registrato al sistema MaaS, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 138	Si controlla che, attraverso una richiesta HTTP, un utente possa autenticarsi al sistema MaaS, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 139	Si controlla che, attraverso una richiesta HTTP, un Super Amministratore possa autenticarsi al sistema MaaS, e che sia possibile ricevere dal server un messaggio di buona riuscita o d'errore.	Non eseguito.
TU - 140	Si controlla che, attraverso una richiesta HTTP, un utente possa disconnettersi dal sistema MaaS, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 141	Si controlla che, attraverso una richiesta HTTP, un Super Amministratore possa disconnettersi dal sistema MaaS, e che sia possibile ricevere dal server un messaggio di buona riuscita dell'operazione.	Non eseguito.
TU - 142	Si verifica che avvenga una chiamata al metodo <code>editUserAvatar()</code> della classe <code>UserWebAPIUtils</code> .	Non eseguito.
TU - 143	Si verifica che avvenga una chiamata al metodo <code>editUserPersonalData()</code> della classe <code>UserWebAPIUtils</code> .	Non eseguito.
TU - 144	Si verifica che avvenga una chiamata al metodo <code>editUserPassword()</code> della classe <code>UserWebAPIUtils</code> .	Non eseguito.
TU - 145	Si verifica che avvenga una chiamata al metodo <code>getUsers()</code> della classe <code>UserWebAPIUtils</code> .	Non eseguito.
TU - 146	Si verifica che avvenga una chiamata al metodo <code>getUser()</code> della classe <code>UserWebAPIUtils</code> .	Non eseguito.
TU - 147	Si verifica che avvenga una chiamata al metodo <code>deleteUser()</code> della classe <code>UserWebAPIUtils</code> .	Non eseguito.



TU - 148	Si verifica che avvenga una chiamata al metodo <b>searchUser()</b> della classe <b>UserWebAPIUtils</b> .	Non eseguito.
TU - 149	Si verifica che avvenga una chiamata al metodo <b>forgotPassword()</b> della classe <b>UserWebAPIUtils</b> .	Non eseguito.
TU - 150	Si verifica che avvenga una chiamata al metodo <b>changeRole()</b> della classe <b>UserWebAPIUtils</b> .	Non eseguito.
TU - 151	Si verifica che avvenga una chiamata al metodo <b>sendInvite()</b> della classe <b>UserWebAPIUtils</b> .	Non eseguito.
TU - 152	Si verifica che venga creata una Action di tipo <b>EDIT_USER_AVATAR_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 153	Si verifica che venga creata una Action di tipo <b>EDIT_USER_PERSONAL_DATA_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 154	Si verifica che venga creata una Action di tipo <b>EDIT_USER_PASSWORD_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 155	Si verifica che venga creata una Action di tipo <b>GET_USERS_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 156	Si verifica che venga creata una Action di tipo <b>GET_USER_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 157	Si verifica che venga creata una Action di tipo <b>DELETE_USER_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 158	Si verifica che venga creata una Action di tipo <b>SEARCH_USER_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 159	Si verifica che venga creata una Action di tipo <b>FORGOT_PASSWORD_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 160	Si verifica che venga creata una Action di tipo <b>CHANGE_ROLE_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 161	Si verifica che venga creata una Action di tipo <b>SEND_INVITE_RESPONSE</b> e che venga inoltrata alla Store <b>UserStore</b> .	Non eseguito.
TU - 162	Si verifica che avvenga una chiamata al metodo <b>getCollections()</b> della classe <b>CollectionWebAPIUtils</b> .	Non eseguito.



TU - 163	Si verifica che avvenga una chiamata al metodo <code>createCollection()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 164	Si verifica che avvenga una chiamata al metodo <code>editCollection()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 165	Si verifica che avvenga una chiamata al metodo <code>deleteCollection()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 166	Si verifica che avvenga una chiamata al metodo <code>retrieveCollectionDSLIS()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 167	Si verifica che avvenga una chiamata al metodo <code>searchCollection()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 168	Si verifica che avvenga una chiamata al metodo <code>executeCollection()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 169	Si verifica che avvenga una chiamata al metodo <code>sendEmail()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 170	Si verifica che avvenga una chiamata al metodo <code>export()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 171	Si verifica che avvenga una chiamata al metodo <code>exportDSLIS()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 172	Si verifica che avvenga una chiamata al metodo <code>importDSLIS()</code> della classe <code>CollectionWebAPIUtils</code> .	Non eseguito.
TU - 173	Si verifica che venga creata una Action di tipo <code>GET_COLLECTIONS_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 174	Si verifica che venga creata una Action di tipo <code>CREATE_COLLECTION_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 175	Si verifica che venga creata una Action di tipo <code>EDIT_COLLECTION_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 176	Si verifica che venga creata una Action di tipo <code>DELETE_COLLECTION_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 177	Si verifica che venga creata una Action di tipo <code>RETRIEVE_COLLECTION_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.



TU - 178	Si verifica che venga creata una Action di tipo <code>SEARCH_COLLECTION_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 179	Si verifica che venga creata una Action di tipo <code>EXECUTE_COLLECTION_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 180	Si verifica che venga creata una Action di tipo <code>SEND_EMAIL_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 181	Si verifica che venga creata una Action di tipo <code>EXPORT_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 182	Si verifica che venga creata una Action di tipo <code>EXPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 183	Si verifica che venga creata una Action di tipo <code>IMPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>CollectionStore</code> .	Non eseguito.
TU - 184	Si verifica che avvenga una chiamata al metodo <code>getDocuments()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 185	Si verifica che avvenga una chiamata al metodo <code>createDocument()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 186	Si verifica che avvenga una chiamata al metodo <code>editDocument()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 187	Si verifica che avvenga una chiamata al metodo <code>deleteDocument()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 188	Si verifica che avvenga una chiamata al metodo <code>searchDocument()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 189	Si verifica che avvenga una chiamata al metodo <code>executeDocument()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 190	Si verifica che avvenga una chiamata al metodo <code>sendEmail()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 191	Si verifica che avvenga una chiamata al metodo <code>exportDSLIS()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 192	Si verifica che avvenga una chiamata al metodo <code>retrieveDocumentDSLIS()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 193	Si verifica che avvenga una chiamata al metodo <code>export()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.



TU - 194	Si verifica che avvenga una chiamata al metodo <code>importDSLIS()</code> della classe <code>DocumentWebAPIUtils</code> .	Non eseguito.
TU - 195	Si verifica che venga creata una Action di tipo <code>GET_DOCUMENT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 196	Si verifica che venga creata una Action di tipo <code>CREATE_DOCUMENT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 197	Si verifica che venga creata una Action di tipo <code>EDIT_DOCUMENT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 198	Si verifica che venga creata una Action di tipo <code>DELETE_DOCUMENT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 199	Si verifica che venga creata una Action di tipo <code>SEARCH_DOCUMENT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 200	Si verifica che venga creata una Action di tipo <code>EXECUTE_DOCUMENT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 201	Si verifica che venga creata una Action di tipo <code>SEND_EMAIL_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 202	Si verifica che venga creata una Action di tipo <code>EXPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 203	Si verifica che venga creata una Action di tipo <code>RETRIEVE_DOCUMENT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 204	Si verifica che venga creata una Action di tipo <code>EXPORT_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 205	Si verifica che venga creata una Action di tipo <code>IMPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>DocumentStore</code> .	Non eseguito.
TU - 206	Si verifica che avvenga una chiamata al metodo <code>getDashboards()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 207	Si verifica che avvenga una chiamata al metodo <code>createDashboard()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.



TU - 208	Si verifica che avvenga una chiamata al metodo <code>editDashboard()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 209	Si verifica che avvenga una chiamata al metodo <code>deleteDashboard()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 210	Si verifica che avvenga una chiamata al metodo <code>searchDashboard()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 211	Si verifica che avvenga una chiamata al metodo <code>executeDashboard()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 212	Si verifica che avvenga una chiamata al metodo <code>exportDSLIS()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 213	Si verifica che avvenga una chiamata al metodo <code>retrieveDashboardDSLIS()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 214	Si verifica che avvenga una chiamata al metodo <code>importDSLIS()</code> della classe <code>DashboardWebAPIUtils</code> .	Non eseguito.
TU - 215	Si verifica che venga creata una Action di tipo <code>GET_DASHBOARDS_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 216	Si verifica che venga creata una Action di tipo <code>CREATE_DASHBOARD_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 217	Si verifica che venga creata una Action di tipo <code>EDIT_DASHBOARD_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 218	Si verifica che venga creata una Action di tipo <code>DELETE_DASHBOARD_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 219	Si verifica che venga creata una Action di tipo <code>SEARCH_DASHBOARD_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 220	Si verifica che venga creata una Action di tipo <code>EXECUTE_DASHBOARD_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 221	Si verifica che venga creata una Action di tipo <code>EXPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 222	Si verifica che venga creata una Action di tipo <code>RETRIEVE_DASHBOARD_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.



TU - 223	Si verifica che venga creata una Action di tipo <code>IMPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>DashboardStore</code> .	Non eseguito.
TU - 224	Si verifica che avvenga una chiamata al metodo <code>getCell()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 225	Si verifica che avvenga una chiamata al metodo <code>createCell()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 226	Si verifica che avvenga una chiamata al metodo <code>editCell()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 227	Si verifica che avvenga una chiamata al metodo <code>deleteCell()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 228	Si verifica che avvenga una chiamata al metodo <code>searchCell()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 229	Si verifica che avvenga una chiamata al metodo <code>executeCell()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 230	Si verifica che avvenga una chiamata al metodo <code>exportDSLIS()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 231	Si verifica che avvenga una chiamata al metodo <code>retrieveCellDSLIS()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 232	Si verifica che avvenga una chiamata al metodo <code>importDSLIS()</code> della classe <code>CellWebAPIUtils</code> .	Non eseguito.
TU - 233	Si verifica che venga creata una Action di tipo <code>GET_CELL_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 234	Si verifica che venga creata una Action di tipo <code>CREATE_CELL_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 235	Si verifica che venga creata una Action di tipo <code>EDIT_CELL_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 236	Si verifica che venga creata una Action di tipo <code>DELETE_CELL_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 237	Si verifica che venga creata una Action di tipo <code>SEARCH_CELL_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 238	Si verifica che venga creata una Action di tipo <code>EXECUTE_CELL_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.



TU - 239	Si verifica che venga creata una Action di tipo <code>EXPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 240	Si verifica che venga creata una Action di tipo <code>RETRIEVE_CELL_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 241	Si verifica che venga creata una Action di tipo <code>IMPORT_DSLIS_RESPONSE</code> e che venga inoltrata alla Store <code>CellStore</code> .	Non eseguito.
TU - 242	Si verifica che avvenga una chiamata al metodo <code>createExternalDatabase()</code> della classe <code>ExternalDatabaseWebAPIUtils</code> .	Non eseguito.
TU - 243	Si verifica che avvenga una chiamata al metodo <code>deleteExternalDatabase()</code> della classe <code>ExternalDatabaseWebAPIUtils</code> .	Non eseguito.
TU - 244	Si verifica che avvenga una chiamata al metodo <code>getExternalDatabases()</code> della classe <code>ExternalDatabaseWebAPIUtils</code> .	Non eseguito.
TU - 245	Si verifica che avvenga una chiamata al metodo <code>searchExternalDatabase()</code> della classe <code>ExternalDatabaseWebAPIUtils</code> .	Non eseguito.
TU - 246	Si verifica che avvenga una chiamata al metodo <code>allowExternalDatabaseAccess()</code> della classe <code>ExternalDatabaseWebAPIUtils</code> .	Non eseguito.
TU - 247	Si verifica che avvenga una chiamata al metodo <code>denyExternalDatabaseAccess()</code> della classe <code>ExternalDatabaseWebAPIUtils</code> .	Non eseguito.
TU - 248	Si verifica che venga creata una Action di tipo <code>CREATE_EXTERNAL_DATABASE_RESPONSE</code> e che venga inoltrata alla Store <code>ExternalDatabaseStore</code> .	Non eseguito.
TU - 249	Si verifica che venga creata una Action di tipo <code>DELETE_EXTERNAL_DATABASE_RESPONSE</code> e che venga inoltrata alla Store <code>ExternalDatabaseStore</code> .	Non eseguito.
TU - 250	Si verifica che venga creata una Action di tipo <code>GET_EXTERNAL_DATABASES_RESPONSE</code> e che venga inoltrata alla Store <code>ExternalDatabaseStore</code> .	Non eseguito.
TU - 251	Si verifica che venga creata una Action di tipo <code>SEARCH_EXTERNAL_DATABASE_RESPONSE</code> e che venga inoltrata alla Store <code>ExternalDatabaseStore</code> .	Non eseguito.





TU - 252	Si verifica che venga creata una Action di tipo AL: LOW_EXTERNAL_DATABASE_ACCESS_RESPONSE e che venga inoltrata alla Store ExternalDatabaseStore.	Non eseguito.
TU - 253	Si verifica che venga creata una Action di tipo DE: NY_EXTERNAL_DATABASE_ACCESS_RESPONSE e che venga inoltrata alla Store ExternalDatabaseStore.	Non eseguito.
TU - 254	Si verifica che avvenga una chiamata al metodo impersonateUser() della classe SuperAmministratoreWebAPIUtils.	Non eseguito.
TU - 255	Si verifica che venga creata una Action di tipo IMPER: SONATE_USER_RESPONSE e che venga inoltrata alla Store SuperAmministratoreStore.	Non eseguito.
TU - 256	Si verifica che avvenga una chiamata al metodo getCompanies() della classe CompanyWebAPIUtils.	Non eseguito.
TU - 257	Si verifica che avvenga una chiamata al metodo createCompany() della classe CompanyWebAPIUtils.	Non eseguito.
TU - 258	Si verifica che avvenga una chiamata al metodo getCompany() della classe CompanyWebAPIUtils.	Non eseguito.
TU - 259	Si verifica che avvenga una chiamata al metodo editCompany() della classe CompanyWebAPIUtils.	Non eseguito.
TU - 260	Si verifica che avvenga una chiamata al metodo deleteCompany() della classe CompanyWebAPIUtils.	Non eseguito.
TU - 261	Si verifica che avvenga una chiamata al metodo searchCompany() della classe CompanyWebAPIUtils.	Non eseguito.
TU - 262	Si verifica che venga creata una Action di tipo GET_COMPANIES_RESPONSE e che venga inoltrata alla Store CompanyStore.	Non eseguito.
TU - 263	Si verifica che venga creata una Action di tipo CREA: TE_COMPANY_RESPONSE e che venga inoltrata alla Store CompanyStore.	Non eseguito.
TU - 264	Si verifica che venga creata una Action di tipo GET_COMPANY_RESPONSE e che venga inoltrata alla Store CompanyStore.	Non eseguito.
TU - 265	Si verifica che venga creata una Action di tipo EDIT_COMPANY_RESPONSE e che venga inoltrata alla Store CompanyStore.	Non eseguito.
TU - 266	Si verifica che venga creata una Action di tipo DELE: TE_COMPANY_RESPONSE e che venga inoltrata alla Store CompanyStore.	Non eseguito.



TU - 267	Si verifica che venga creata una Action di tipo SEARCH_COMPANY_RESPONSE e che venga inoltrata alla Store CompanyStore.	Non eseguito.
TU - 268	Si verifica che avvenga una chiamata al metodo createUser() della classe SessionWebAPIUtils.	Non eseguito.
TU - 269	Si verifica che avvenga una chiamata al metodo loginUser() della classe SessionWebAPIUtils.	Non eseguito.
TU - 270	Si verifica che avvenga una chiamata al metodo loginSuperAmministratore() della classe SessionWebAPIUtils.	Non eseguito.
TU - 271	Si verifica che avvenga una chiamata al metodo logoutUser() della classe SessionWebAPIUtils.	Non eseguito.
TU - 272	Si verifica che avvenga una chiamata al metodo logoutSuperAmministratore() della classe SessionWebAPIUtils.	Non eseguito.
TU - 273	Si verifica che venga creata una Action di tipo CREATE_USER_RESPONSE e che venga inoltrata alla Store SessionStore.	Non eseguito.
TU - 274	Si verifica che venga creata una Action di tipo LOGIN_USER_RESPONSE e che venga inoltrata alla Store SessionStore.	Non eseguito.
TU - 275	Si verifica che venga creata una Action di tipo LOGIN_SUPER_AMMINISTRATORE_RESPONSE e che venga inoltrata alla Store SessionStore.	Non eseguito.
TU - 276	Si verifica che venga creata una Action di tipo LOGOUT_USER_RESPONSE e che venga inoltrata alla Store SessionStore.	Non eseguito.
TU - 277	Si verifica che venga creata una Action di tipo LOGOUT_SUPER_AMMINISTRATORE_RESPONSE e che venga inoltrata alla Store SessionStore.	Non eseguito.
TU - 278	Si controlla che vengano preparate Action riguardanti azioni del back-end, senza che avvengano errori.	Non eseguito.
TU - 279	Si controlla che vengano preparate Action riguardanti azioni del front-end, senza che avvengano errori.	Non eseguito.
TU - 280	Si controlla che sia possibile individuare l'origine delle API del front-end, senza che avvengano errori.	Non eseguito.
TU - 281	Si controlla che sia possibile individuare i punti d'arrivo delle API del front-end, senza che avvengano errori.	Non eseguito.



TU - 282	Si controlla che sia possibile individuare le origini dei dati trasmessi, senza che avvengano errori.	Non eseguito.
TU - 283	Si controlla che sia possibile identificare il tipo di Action generata da azioni del front-end, senza che avvengano errori.	Non eseguito.
TU - 284	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 285	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 286	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 287	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 288	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 289	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 290	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 291	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 292	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 293	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 294	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.



TU - 295	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 296	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 297	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 298	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 299	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 300	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 301	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 302	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 303	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 304	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 305	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 306	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 307	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.



TU - 308	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 309	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 310	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 311	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 312	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 313	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 314	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 315	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 316	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 317	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 318	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 319	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 320	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.



TU - 321	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 322	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 323	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 324	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 325	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 326	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 327	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 328	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 329	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 330	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 331	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 332	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 333	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.



TU - 334	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 335	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 336	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 337	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 338	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 339	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 340	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 341	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 342	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 343	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 344	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 345	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 346	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.



TU - 347	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 348	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 349	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 350	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 351	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 352	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 353	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 354	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 355	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 356	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 357	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 358	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 359	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.





TU - 360	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 361	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 362	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 363	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 364	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 365	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 366	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 367	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 368	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 369	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 370	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 371	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 372	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.



TU - 373	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 374	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 375	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 376	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 377	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 378	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 379	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 380	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 381	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 382	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 383	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 384	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 385	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.



TU - 386	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 387	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 388	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 389	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 390	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 391	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 392	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 393	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 394	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 395	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <code>render()</code> , senza che si presentino errori.	Non eseguito.
TU - 396	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 397	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 398	Si verifica che venga richiamato il metodo <code>setState(viewState)</code> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.



TU - 399	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 400	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 401	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 402	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 403	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 404	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 405	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 406	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 407	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 408	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 409	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 410	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 411	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.



TU - 412	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 413	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 414	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 415	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 416	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 417	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 418	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 419	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 420	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 421	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 422	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 423	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 424	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.



TU - 425	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 426	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 427	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 428	Si verifica che venga richiamato il metodo <b>setState(viewState)</b> offerto dalle Component API di React, senza che si presentino errori.	Non eseguito.
TU - 429	Si verifica che si possa ottenere un oggetto figlio di una componente (React View) DOM, nativa o creata appositamente, che ne rappresenti una riproduzione virtuale, senza che si presentino errori.	Non eseguito.
TU - 430	Si verifica che si possa ottenere un oggetto JavaScript che rappresenti lo stato iniziale di una view, senza che si presentino errori.	Non eseguito.
TU - 431	Si verifica che la view venga composta interamente prima dell'esecuzione del metodo <b>render()</b> , senza che si presentino errori.	Non eseguito.
TU - 432	Si verifica che si possano manipolare le DOM ed ottenere dati da esse, senza che si presentino errori.	Non eseguito.
TU - 433	Si verifica che venga effettuata una pulizia generale prima della deallocazione del componente, senza che si presentino errori.	Non eseguito.
TU - 434	Si verifica che il metodo faccia da interfaccia a tutti i metodi omonimi delle classi della view.	Non eseguito.
TU - 435	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 436	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 437	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 438	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.



TU - 439	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 440	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 441	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 442	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 443	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 444	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 445	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 446	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 447	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 448	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 449	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 450	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 451	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.



TU - 452	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 453	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 454	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 455	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 456	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 457	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 458	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 459	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 460	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 461	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 462	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 463	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 464	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.





TU - 465	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 466	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 467	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 468	Si verifica che si possa emettere un segnale indicante un cambiamento nella corrispondente Store, senza che avvengano errori.	Non eseguito.
TU - 469	Si verifica che si possa aggiungere alla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 470	Si verifica che si possa togliere dalla Store una callback in ascolto di un segnale di cambiamento specifico, senza che avvengano errori.	Non eseguito.
TU - 471	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>RegistrazioneView</b> , senza che avvengano errori.	Non eseguito.
TU - 472	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>LoginView</b> , senza che avvengano errori.	Non eseguito.
TU - 473	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>RecuperoPasswordView</b> , senza che avvengano errori.	Non eseguito.
TU - 474	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>GestioneDSLISView</b> , senza che avvengano errori.	Non eseguito.
TU - 475	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>GestioneDatabaseEsterniView</b> , senza che avvengano errori.	Non eseguito.
TU - 476	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>ModificaPermessiAccessoDatabaseEsterniView</b> , senza che avvengano errori.	Non eseguito.



TU - 477	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>RegistrazioneCollaboratoreView</b> , senza che avvengano errori.	Non eseguito.
TU - 478	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>CreazioneDSLISView</b> , senza che avvengano errori.	Non eseguito.
TU - 479	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>ModificaDSLISView</b> , senza che avvengano errori.	Non eseguito.
TU - 480	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>DashboardView</b> , senza che avvengano errori.	Non eseguito.
TU - 481	Si verifica che il metodo restituisca un ogget: to JavaScript, riferimento del file corrispondente alla view <b>ConfermaRegistrazioneView</b> , senza che avvengano errori.	Non eseguito.
TU - 482	Si verifica che il metodo restituisca un ogget: to JavaScript, riferimento del file corrispondente alla view <b>GestionePermessiDSLISView</b> , senza che avvengano errori.	Non eseguito.
TU - 483	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>ImpostazioniApplicazioneView</b> , senza che avvengano errori.	Non eseguito.
TU - 484	Si verifica che il metodo restituisca un ogget: to JavaScript, riferimento del file corrispondente alla view <b>GestioneUtentiAziendaView</b> , senza che avvengano errori.	Non eseguito.
TU - 485	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>ModificaDashboardPredefinitaView</b> , senza che avvengano errori.	Non eseguito.
TU - 486	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>ModificaPasswordView</b> , senza che avvengano errori.	Non eseguito.
TU - 487	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>VisualizzazioneDSLISView</b> , senza che avvengano errori.	Non eseguito.



TU - 488	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>ModificaDatiAnagraficiView</b> , senza che avvengano errori.	Non eseguito.
TU - 489	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>GestioneDatiAziendeView</b> , senza che avvengano errori.	Non eseguito.
TU - 490	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>SuperAmministratoreDashboardView</b> , senza che avvengano errori.	Non eseguito.
TU - 491	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>GestioneDatiUtenteView</b> , senza che avvengano errori.	Non eseguito.
TU - 492	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>DocumentView</b> , senza che avvengano errori.	Non eseguito.
TU - 493	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>DSLISListView</b> , senza che avvengano errori.	Non eseguito.
TU - 494	Si verifica che il metodo restituisca un oggetto Ja: vaScript, riferimento del file corrispondente alla view <b>CollectionView</b> , senza che avvengano errori.	Non eseguito.
TU - 495	Si verifica che il metodo restituisca un oggetto JavaScript, riferimento del file corrispondente alla view <b>CellView</b> , senza che avvengano errori.	Non eseguito.
TU - 496	Si verifica che il metodo restituisca un oggetto JavaScript rappresentante lo stato della Store per la memorizzazione degli osservatori (le view), senza che avvengano errori.	Non eseguito.

Tabella A.4: Test di Unità



## B Resoconto delle attività di verifica

### B.1 Riassunto delle attività di verifica

#### B.1.1 Revisione dei Requisiti

L'attività di verifica svolta dai Verificatori è avvenuta come determinato dal *Piano di Progetto v3.0.0* al termine della stesura di ogni documento previsto. La verifica svolta sui documenti e sui processi è avvenuta seguendo le indicazioni delle *Norme di Progetto v3.0.0* e misurando le metriche indicate in 2.7.1.

#### B.1.2 Revisione di Progettazione

L'attività di verifica svolta dai *Verificatori* è avvenuta come determinato dal *Piano di Progetto v3.0.0* al termine della stesura di ogni documento previsto. La verifica svolta sui documenti è avvenuta seguendo le indicazioni delle *Norme di Progetto v3.0.0* e misurando le metriche indicate in 2.7.1. Successivamente si è proceduto con le misurazioni delle metriche relative ai documenti. Sono quindi state effettuate le misurazioni mediante le metriche sui processi per valutarne la bontà e fornire una base per la pianificazione dei cicli *PDCA<sub>G</sub>*.

### B.2 Dettaglio delle verifiche tramite analisi

#### B.2.1 Analisi

##### B.2.1.1 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante l'analisi e relativo esito basato sui range stabiliti in 2.7.1.

Documento	Valore indice	Esito
<i>Analisi dei Requisiti v1.0.0</i>	54	superato
<i>Glossario v1.0.0</i>	51	superato
<i>Norme di Progetto v1.0.0</i>	55	superato
<i>Piano di Progetto v1.0.0</i>	60	superato
<i>Piano di Qualifica v1.0.0</i>	57	superato
<i>Studio di Fattibilità v1.0.0</i>	50	superato

Tabella A.5: Esiti verifica documenti in fase di Analisi

#### B.2.2 Progettazione

##### B.2.2.1 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante la progettazione architeturale e relativo esito basato sui range stabiliti in ??.



Documento	Valore indice	Esito
<i>Analisi dei Requisiti v3.0.0</i>	51	superato
<i>Glossario v3.0.0</i>	49	superato
<i>Norme di Progetto v3.0.0</i>	47	superato
<i>Piano di Progetto v3.0.0</i>	73	superato
<i>Piano di Qualifica v3.0.0</i>	50	superato
<i>Definizione di Prodotto v3.0.0</i>	58	superato

Tabella A.6: Esiti verifica documenti, Progettazione Architettuale

## B.3 Dettaglio dell'esito delle revisioni

Lo sviluppo di questo progetto didattico si basa sull'attraversamento di quattro revisioni presiedute dal committente. Tre delle quattro revisioni produrranno delle segnalazioni degli errori riscontrati da parte del committente, deve seguire un report di come sono state risolte in ogni documento.

### B.3.1 Revisione dei Requisiti

Per la Revisione dei Requisiti le segnalazioni da parte del committente sono state corrette:

- *Norme di Progetto*: il documento non ha subito modifiche in quanto considerato "eccellente".
- *Analisi dei Requisiti*: le *Norme di Progetto v3.0.0* descrivono la modalità di consegna che è stata ben definita includendo la generazione dei nomi dei documenti con la relativa versione. Inoltre sono stati rivisti e ampliati tutti i requisiti e casi d'uso segnalati dal committente;
- *Piano di Progetto*: sono stati approfonditi i rischi evidenziando i riscontri. Inoltre sono state rinominate le "fasi" raggiungendo una denominazione più significativa sugli stati di avanzamento.
- *Piano di Qualifica*: sono stati spostati i contenuti discorsivi, relativi alla qualità, in appendice. Sono stati definiti i test di validazione, sistema e integrazione ottenendo così una migliore correlazione con quanto specificato nelle Norme di Progetto.
- *Glossario*: il documento non ha subito modifiche in quanto considerato positivo.



## C Qualità

La qualità perseguita nel presente documento si basa sugli standard ISO/IEC 15504 e ISO/IEC 9126 con l'obiettivo di approfondirne incrementalmente la copertura.

### C.1 Qualità di processo

La qualità del processo è un fattore determinante per la qualità del prodotto. Si è deciso di perseguirla servendosi dei modelli  $SPICE_G$  e  $PDCA_G$ .

#### C.1.1 Standard ISO/IEC 15504

La qualità di processo definita in questo standard come SPICE (Software Process Improvement and Capability Determination), specifica come la qualità è collegata alla maturazione dei processi. Il  $team_G$  ha scelto questo standard ai fini di una valutazione oggettiva dei processi, per darne un giudizio di maturità e per individuare azioni migliorative. Vengono individuati dei livelli di maturità al quale il fornitore può fare riferimento per determinare le proprie capacità organizzative. Vengono definiti:

- **Modelli di riferimento:**

- Dimensione del processo;
- Livelli di maturità dei processi:
  - \* **5:** Ottimizzato
  - \* **4:** Predicibile
  - \* **3:** Stabilito
  - \* **2:** Gestito
  - \* **1:** Eseguito
  - \* **0:** Incompleto

La capacità di un processo viene misurata tramite degli attributi che sono assimilabili alle metriche dei processi individuate in 2.7.2, in particolare la  $Schedule\ Variance_G$  permette di capire se un processo è incompleto o gestito; il gruppo raggiungerà uno stato accettabile quando i processi diventeranno predicibili ossia quando la Schedule Variance subirà al più lievi oscillazioni;

- **Stime:** si concretizzano in una struttura per la misurazione composta da:
  - I processi di misurazione, indicati nel *Piano di Progetto v3.0.0*;
  - Un modello per la misurazione identificabile in questo documento;
  - Gli strumenti utilizzati, specificati nelle *Norme di Progetto v3.0.0*.
- **Competenze e Qualifiche di chi controlla:** lo standard redige in modo rigoroso una serie di attività volte a formare chi opera l'attività di verifica e di stesura del Piano di Qualifica. Tali competenze sono assenti all'interno del gruppo e, considerato

che effettuare una formazione in linea con quanto specificato dallo standard sarebbe impossibile, tutti i membri si impegnano a studiare ed applicare al meglio quanto descritto in questo documento.

### C.1.2 Ciclo di Deming

La qualità va ricercata non sul prodotto bensì sui processi alla base del prodotto, per questo il team ha scelto il metodo PDCA per il controllo delle attività di processo ripetibili e misurabili e per la manutenibilità dei processi stessi. Esso prevede l'iterazione ripetuta tra i quattro stadi definiti di seguito, assicurando un incremento della qualità ad ogni ciclo.

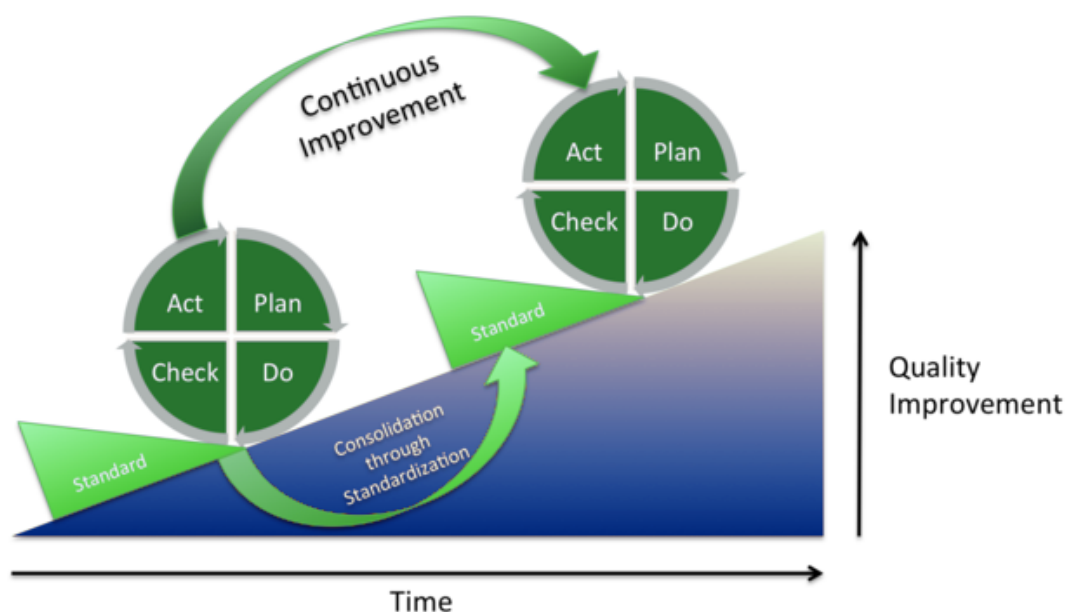


Figura 4: Continuous quality improvement with PDCA

1. **PLAN:** vengono stabiliti gli obiettivi e i processi di miglioramento necessari per raggiungere la qualità attesa, nel dettaglio:
  - Identificare il problema, o i processi da migliorare; per descrivere il problema è necessario raccogliere i dati tramite misurazioni;
  - Analizzare il problema e individuare gli effetti negativi, definendo la loro importanza e le priorità di intervento;
  - Definire gli obiettivi di massima in modo chiaro e quantitativo, indicando i benefici ottenibili con il suo raggiungimento. Devono essere definiti anche i tempi, gli indicatori e gli strumenti di controllo.
2. **DO:** viene implementato il punto precedente, applicando le soluzioni individuate al problema;



3. **CHECK:** verificare i risultati delle azioni intraprese; un confronto con i risultati attesi sarà il riscontro se quanto operato va nella direzione giusta. Vanno considerate metriche come la Schedule Variance (vedi 2.7.2.1) e la completezza dei risultati attesi soddisfatti, vanno elaborati grafici e tabelle per avere una visione chiara di quanto rilevato. Se si è raggiunto l'obiettivo definito nello stadio di Plan, si può passare allo stadio di Act, altrimenti è necessario ripetere un nuovo ciclo PDCA sullo stesso problema, analizzando i vari stadi del ciclo precedente individuandone le cause del non raggiungimento dell'obiettivo stabilito;
4. **ACT:** La soluzione individuata viene standardizzata e tutti i membri del gruppo vengono informati e formati. Si potrà eseguire tramite riunioni o strumenti di messaggistica interna al gruppo. Terminato questo stadio si procederà con una nuova iterazione a partire dal punto 1.

## C.2 Qualità di prodotto

Sono necessari degli obiettivi rivolti direttamente alla qualità del prodotto per massimizzare l'efficacia. Lo standard ISO/IEC 9126 classifica la qualità del software e definisce delle metriche per la sua misurazione.

### C.2.1 Standard ISO/IEC 9126

Prendendo come riferimento questo standard il team MINT si impegna a garantire nel prodotto MaaS le qualità da esso definite.

Esso si suddivide in:

- **Quality model:** classifica la qualità del software in un set di caratteristiche che verranno approfondite nel corso del progetto:
  - Functionality: viene controllata grazie al tracciamento dei requisiti individuati ed analizzati e i componenti;
  - Reliability: viene dimostrata combinando i test;
  - Usability: viene controllata con i test di validazione, inoltre la stesura del manuale d'uso aiuterà a verificarne l'usabilità e ad intervenire laddove necessario;
  - Efficiency: combinando analisi statica e dinamica controlliamo che il prodotto sia efficiente;
  - Maintainability: viene realizzata con l'utilizzo di design pattern e la stesura di documentazione dettagliata;
  - Portability: essendo  $MaaS_G$  un applicazione Web non ci sono particolari problemi di portabilità per gli utenti.
- **External metrics:** sono le metriche rilevate tramite analisi dinamica specificate in 2.7;
- **Internal metrics:** sono le metriche rilevate in analisi statica specificate in 2.7;
- **Quality in use metrics:** si tratta di metriche rilevabili allo stato di prodotto *usabile* in condizioni reali, si rimanda la definizione di tale aspetto a quando verranno trattate





le considerazioni sull'usabilità del prodotto in uno scenario di utilizzo reale, questo deve avvenire non oltre la *Progettazione di Dettaglio e Codifica*.



## D PDCA

In questo capitolo verrà descritto come è stato applicato il modello  $PDCA_G$  descritto in questo documento in C.1.2.

### D.1 Revisione dei requisiti

In questo periodo è stata svolta un attività di  $walkthrough_G$  non avendo i dati necessari per effettuare attività di  $inspection_G$ . Non è stato possibile eseguire nessun ciclo  $PDCA_G$  in mancanza di misurazioni sui processi, non avendo quindi modo di pianificare miglioramenti per la qualità, ma è stato studiato e verrà attuato a partire dalla prossima  $milestone_G$ .

### D.2 Revisione di progettazione

**PLAN** Al fine di valutare su quali processi pianificare dei miglioramenti sono state eseguite diverse misurazioni utilizzando le metriche per i processi descritte in 2.7.2.

I risultati ottenuti sono riportati nella seguente tabella:

Metrica	Valore indice	Esito
Produttività di documentazione	186	Superato
Impegno	0,78	Superato
Modifiche	24	Non superato

Tabella A.7: Risultati metriche per i processi, Revisione di progettazione

Analizzando tali dati si è deciso di pianificare le seguenti attività per il miglioramento della qualità dei processi:

Un numero di modifiche troppo elevato grava pesantemente sulla produttività. C'è quindi il bisogno di ridurre tale valore al fine di garantire maggiore produttività e di conseguenza diminuire i costi. Con questo primo ciclo  $PDCA_G$  il gruppo si prefigge dunque l'obiettivo di portare entro un range di accettazione (Vedi 2.7.2) il numero di modifiche approvate.

Probabilmente il numero elevato di modifiche è causato dall'inesperienza del gruppo nel primo periodo, in conseguenza all'aumentare delle conoscenze il numero di modifiche andrà calando ragionevolmente.

In ogni caso, si pianifica di:

- Frammentare maggiormente i task assegnati in sotto-task;
- Specificare in modo esteso cosa prevede ogni singolo sotto-task, escludendo quindi dubbi che poi porteranno a successive richieste di modifica;

**CHECK** Al fine di valutare se le azioni pianificate hanno portato ad un miglioramento dei processi sono state eseguite le necessarie misurazioni.

I risultati ottenuti sono riportati nella seguente tabella:



<b>Metrica</b>	<b>Valore indice</b>	<b>Esito</b>
Produttività di documentazione	121	Superato
Impegno	1,83	Superato
Modifiche	16	Superato

Tabella A.8: Risultati metriche per i processi, Revisione di progettazione

Gli obiettivi posti nello stadio di pianificazione sono stati soddisfatti, si passerà dunque allo stadio di standardizzazione delle soluzioni applicate.