



# Piano di Qualifica

## Informazioni sul documento

<b>Nome documento</b>	Piano di Qualifica
<b>Versione</b>	v4.0.0
<b>Data redazione</b>	2013-12-09
<b>Redattori</b>	<ul style="list-style-type: none"><li>• Adami Alberto</li><li>• Bissacco Nicolò</li><li>• Magnabosco Nicola</li></ul>
<b>Verificatori</b>	<ul style="list-style-type: none"><li>• Scapin Davide</li></ul>
<b>Approvazione</b>	<ul style="list-style-type: none"><li>• Feltre Beatrice</li><li>• <i>Seven Monkeys</i></li></ul>
<b>Lista distribuzione</b>	<ul style="list-style-type: none"><li>• <i>Prof. Tullio Vardanega</i></li><li>• <i>Prof. Riccardo Cardin</i></li><li>• <i>Dott.ssa Gaia Rizzo - DEI</i></li></ul>
<b>Uso</b>	Esterno

## Sommario

Documento riguardante le strategie di verifica e validazione adottate dal gruppo *Seven Monkeys* volte al soddisfacimento dei requisiti qualitativi nello svolgimento del progetto Romeo

## Diario delle Modifiche

Modifica	Autore & Ruolo	Data	Versione
<i>Approvazione documento</i>	Luisetto Luca <i>Responsabile di Progetto</i>	2014-06-03	v4.0.0
<i>Eseguita verifica del documento</i>	Scapin Davide <i>Verificatore</i>	2014-05-30	v3.2.0
<i>Apportate modifiche a seguito della verifica</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-05-29	v3.1.1
<i>Eseguita verifica del documento</i>	Scapin Davide <i>Verificatore</i>	2014-05-26	v3.1.0
<i>Aggiornato lo stato dei test di sistema</i>	Adami Alberto <i>Verificatore Capo</i>	2014-05-22	v3.0.5
<i>Stesura dell'appendice riguardante pianificazione ed esecuzione del collaudo</i>	Adami Alberto <i>Verificatore Capo</i>	2014-05-20	v3.0.4
<i>Inseriti i risultati delle metriche riguardanti il software e i documenti</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-05-17	v3.0.3
<i>Aggiornata appendice C con dettaglio della revisione di qualifica e relative correzioni eseguite</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-04-27	v3.0.2
<i>Apportate modifiche a seguito della revisione di qualifica in particolare: rimosso il capitolo riguardante la gestione amministrativa, esplicitati gli obiettivi riguardanti le metriche, rimosse le tecniche di analisi</i>	Adami Alberto <i>Verificatore Capo</i>	2014-04-23	v3.0.1
<i>Approvazione documento</i>	Scapin Davide <i>Responsabile di Progetto</i>	2014-03-28	v3.0.0
<i>Eseguita verifica del documento</i>	Adami Alberto <i>Verificatore</i>	2014-03-28	v2.2.0
<i>Apportate modifiche a seguito della verifica</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-03-27	v2.1.1
<i>Eseguita verifica del documento</i>	Adami Alberto <i>Verificatore</i>	2014-03-27	v2.1.0
<i>Aggiunto tracciamento test di unità indicando lo stato del test, ovvero se ha avuto successo o se N.I.</i>	Bissacco Nicolò <i>Verificatore Capo</i>	2014-03-26	v2.0.5
<i>Aggiunta sezione riguardante la verifica per quello che concerne l'analisi dei processi (BV e SV) e dei documenti(indice gulpease) per la Fase D</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-03-09	v2.0.4
<i>Aggiornata appendice C con dettaglio della revisione di progettazione e relative correzioni eseguite</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-02-25	v2.0.3

<i>Apportate modifiche in seguito alla revisione di progettazione: Aggiunte metriche per i processi (BV e SV), indicato per ogni processo il livello raggiunto da essi, in ogni fase e obiettivi da perseguire</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-02-20	v2.0.2
<i>aggiunte metriche per il software: copertura del codice</i>	Feltre Beatrice <i>Verificatore Capo</i>	2014-02-14	v2.0.1
<i>Approvazione documento</i>	Adami Alberto <i>Responsabile di Progetto</i>	2013-02-04	v2.0.0
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2014-02-04	v1.2.0
<i>Apportate modifiche a seguito della verifica</i>	Feltre Beatrice <i>Amministratore di Progetto</i>	2014-02-04	v1.1.1
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2014-02-03	v1.1.0
<i>Aggiunto indice gulpease dei documenti</i>	Bissacco Nicolò <i>Verificatore</i>	2014-02-03	v1.0.6
<i>Incremento sezione Pianificazione test</i>	Bissacco Nicolò <i>Verificatore capo</i>	2013-01-24	v1.0.5
<i>Incremento sezione Misure e metriche di qualità aggiungendo accoppiamento afferente ed efferente</i>	Bissacco Nicolò <i>Verificatore capo</i>	2013-01-22	v1.0.4
<i>Aggiunta sezione Pianificazione test</i>	Scapin Davide <i>Verificatore capo</i>	2013-01-14	v1.0.3
<i>Aggiunta sezione Definizione obiettivi</i>	Magnabosco Nicola <i>Responsabile di Progetto</i>	2014-01-13	v1.0.2
<i>Apportate modifiche alla struttura del documento a seguito dei suggerimenti del Committente</i>	Bissacco Nicolò <i>Verificatore Capo</i>	2014-01-10	v1.0.1
<i>Approvazione documento</i>	Feltre Beatrice <i>Responsabile di Progetto</i>	2013-12-19	v1.0.0
<i>Eseguita verifica del documento</i>	Scapin Davide <i>Verificatore</i>	2013-12-19	v0.2.0
<i>Aggiunto indice gulpease dei documenti</i>	Magnabosco Nicola <i>Verificatore Capo</i>	2013-12-19	v0.1.2
<i>Apportate modifiche a seguito della verifica</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-17	v0.1.1
<i>Eseguita verifica del documento</i>	Scapin Davide <i>Verificatore</i>	2013-12-16	v0.1.0
<i>Completata stesura del documento: aggiunta sezione Standard ISO/IEC 9126</i>	Adami Alberto <i>Amministratore di Progetto</i>	2013-12-14	v0.0.6
<i>Aggiunti appendici Resoconto delle attività di verifica e Standard di qualità ISO/IEG 15507</i>	Magnabosco Nicola <i>Verificatore Capo</i>	2013-12-14	v0.0.5
<i>Aggiunta sezione Gestione amministrativa della revisione e relative sottosezioni</i>	Bissacco Nicolò <i>Amministratore di Progetto</i>	2013-12-13	v0.0.4

---

<i>Completata stesura sezione Visione generale della strategia di verifica: aggiunta sottosezione Tecniche, misure e metriche</i>	Bissacco Nicolò <i>Amministratore di Progetto</i>	2013-12-12	v0.0.3
<i>Inizio stesura sezione Visione generale della strategia di verifica: aggiunte sottosezioni Organizzazione, Pianificazione Strategica e temporale, Responsabilità</i>	Magnabosco Nicola <i>Verificatore Capo</i>	2013-12-11	v0.0.2
<i>Inizio stesura del documento. Aggiunta sezione Introduzione</i>	Magnabosco Nicola <i>Verificatore Capo</i>	2013-12-09	v0.0.1

---

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del Documento	1
1.2	Scopo del Prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Normativi	1
1.4.2	Informativi	1
<b>2</b>	<b>Visione generale della strategia di verifica</b>	<b>2</b>
2.1	Definizione obiettivi	2
2.1.1	Qualità di processo	2
2.1.2	Qualità di prodotto	3
2.2	Organizzazione	3
2.3	Pianificazione strategica e temporale	4
2.4	Responsabilità	4
2.5	Risorse necessarie	4
2.6	Tecniche, misure e metriche	5
2.6.1	Tecniche di analisi	5
2.6.2	Misure e metriche di qualità	5
<b>A</b>	<b>Standard di qualità</b>	<b>8</b>
A.1	Standard ISO/IEC 15504	8
A.2	Standard ISO/IEC 9126	9
<b>B</b>	<b>Resoconto delle attività di verifica</b>	<b>11</b>
B.1	Revisione dei Requisiti	11
B.1.1	Documenti	11
B.2	Revisione di Progettazione	12
B.2.1	Documenti	12
B.3	Revisione di Qualifica	12
B.3.1	Documenti	12
B.3.2	Progettazione	12
B.3.3	Codice	13
B.4	Revisione di Accettazione	15
B.4.1	Documenti	15
B.4.2	Codice	15
B.4.3	Verifica della copertura dei requisiti	16
<b>C</b>	<b>Dettaglio dell'esito delle revisioni</b>	<b>17</b>
C.1	Revisione dei Requisiti	17
C.2	Revisione di Progettazione	17
C.3	Revisione di Qualifica	18
<b>D</b>	<b>Pianificazione dei test</b>	<b>19</b>
D.1	Test di sistema	19
D.1.1	Descrizione test di sistema	20
D.2	Test di integrazione	21
D.3	Test di validazione	24
D.3.1	Test TV1	24

D.3.2	Test TV2 . . . . .	24
D.3.3	Test TV3 . . . . .	24
D.3.4	Test TV4 . . . . .	24
D.3.5	Test TV5 . . . . .	24
D.3.6	Test TV6 . . . . .	25
D.3.7	Test TV7 . . . . .	25
D.3.8	Test TV8 . . . . .	25
D.3.9	Test TV9 . . . . .	25
D.3.10	Test TV10 . . . . .	25
D.3.11	Test TV11 . . . . .	26
D.3.12	Test TV12 . . . . .	26
D.4	Test di unità . . . . .	27
<b>E</b>	<b>Pianificazione ed esecuzione del collaudo . . . . .</b>	<b>37</b>

## Elenco delle tabelle

3	Quantificazione degli obiettivi . . . . .	7
4	Indice Gulpease dei documenti presentati in Revisione dei Requisiti .	11
5	Indice Gulpease dei documenti presentati in Revisione di Progettazione	12
6	Indice Gulpease dei documenti presentati in Revisione di Qualifica .	12
7	Indice Gulpease dei documenti presentati in Revisione di Accettazione	15
8	Riepilogo copertura dei requisiti . . . . .	16
9	Tracciamento test di sistema-requisiti . . . . .	20
10	Test di integrazione . . . . .	22
11	Tracciamento componente-test di integrazione . . . . .	23
12	Descrizione test di unità . . . . .	36

## Elenco delle figure

1	Diagramma informale della strategia di integrazione . . . . .	21
---	---	----



# 1 Introduzione

## 1.1 Scopo del Documento

Il *Piano di Qualifica* ha lo scopo di descrivere le strategie che il gruppo intende adottare, al fine di perseguire gli obiettivi qualitativi che si intendono applicare al prodotto da sviluppare. Per soddisfare tali obiettivi, è quindi necessario un processo di verifica costante sulle attività svolte, permettendo così di rilevare eventuali incongruenze e anomalie che si potrebbero riscontrare.

## 1.2 Scopo del Prodotto

Il prodotto che si intende realizzare, denominato Romeo, si propone di fornire un sistema software per applicare la cluster analysis<sub>G</sub> ad immagini biomediche. Lo scopo principale è quello di offrire alla comunità scientifica internazionale uno strumento semplice, ma allo stesso tempo completo e flessibile per applicare gli algoritmi della cluster analysis<sub>G</sub>.

## 1.3 Glossario

Al fine di evitare ogni ambiguità e per permettere al lettore una migliore comprensione dei termini e acronimi utilizzati nei vari documenti formali, essi sono riportati nel *Glossario v3.0.0* che contiene una descrizione approfondita di tali termini e acronimi.

Ogni volta che compare un termine presente nel *Glossario*, esso è marcato con una “G” in pedice.

## 1.4 Riferimenti

### 1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v4.0.0*;
- **Capitolato d'appalto C3:** Romeo, Medical Image Cluster Analysis Tool <http://www.math.unipd.it/~tullio/IS-1/2013/Progetto/C3p.pdf>
- **Standard ISO<sub>G</sub>/IEC<sub>G</sub>9126:** Software Engineering - Product Quality [http://en.wikipedia.org/wiki/ISO/IEC\\_9126](http://en.wikipedia.org/wiki/ISO/IEC_9126)
- **Standard ISO<sub>G</sub>/IEC<sub>G</sub>15504 (SPICE):** Information Technology — Process assessment
  - [http://en.wikipedia.org/wiki/ISO/IEC\\_15504](http://en.wikipedia.org/wiki/ISO/IEC_15504);
  - <http://www.ehealthinformation.ca/documents/1072.pdf>

### 1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v4.0.0*;
- **Slide dei corsi di Ingegneria del Software modulo A:** <http://www.math.unipd.it/~tullio/IS-1/2013>;
- **SWEBOK V3 (2004):** Chapter 11 - Software Quality <http://www.computer.org/portal/web/swbok/html/ch11>;
- **Indice Gulpease:** [http://it.wikipedia.org/wiki/Indice\\_Gulpease](http://it.wikipedia.org/wiki/Indice_Gulpease).

## 2 Visione generale della strategia di verifica

### 2.1 Definizione obiettivi

Al fine di garantire la qualità del prodotto da realizzare, è necessario definire a priori gli obiettivi da raggiungere. Senza una guida che aiuti a delineare gli obiettivi, il concetto stesso di qualità rimane astratto. Per questo il gruppo *Seven Monkeys*, intende adottare degli standard internazionali che forniscono un modello per la *qualità di processo* e la *qualità di prodotto*.

#### 2.1.1 Qualità di processo

Per ottenere un prodotto di qualità, è necessario che questo derivi da un processo di qualità. Lo standard adottato per garantire la qualità dei processi è l' ISO<sub>G</sub>/IEC<sub>G</sub> 15504<sup>1</sup>, che definisce un modello e delle metriche per la valutazione della maturità dei processi. Di seguito verranno illustrati i processi individuati (descritti nel documento *Norme di Progetto v3.0.0*) e i relativi obiettivi di qualità che il gruppo *Seven Monkeys* si prefiggerà di raggiungere per ognuno di essi in relazione alla scala di misura fornita dallo standard e dettagliatamente descritta in appendice A.1.

#### Processo di Sviluppo

##### Obiettivo:

- livello: 4;
- process Measurement: Largely achieved;
- process Control: Partially achieved.

#### Processo di Documentazione

##### Obiettivo:

- livello: 5;
- process Change: Fully achieved;
- process Continuous Improvement: Fully achieved.

#### Processo di Organizzazione dell'infrastruttura

##### Obiettivo:

- livello: 3;
- process Definition: Fully achieved;
- process Deployment: Fully achieved.

#### Processo di Gestione del Progetto

##### Obiettivo:

- livello: 3;
- process Definition: Fully achieved;
- process Deployment: Partially achieved.

---

<sup>1</sup>Per una descrizione dettagliata dello standard, consultare l'appendice A.1

### Processo di Garanzia della Qualità

#### Obiettivo:

- livello: 5;
- process Change: Partially achieved;
- process Continuous Improvement: Partially achieved.

### Processo di Verifica

#### Obiettivo:

- livello: 4;
- process Measurement: Fully achieved;
- process Control: Partially achieved.

#### 2.1.2 Qualità di prodotto

Per garantire la qualità del prodotto software da realizzare, il gruppo intende adottare lo standard ISO<sub>G</sub>/IEC<sub>G</sub> 9126<sup>2</sup>. Lo standard suddivide in quattro parti le normative tecniche relative alla qualità del software:

- modello per la *qualità del software*;
- metriche per la *qualità esterna*;
- metriche per la *qualità interna*;
- metriche per la *qualità in uso*.

Oltre che al soddisfacimento delle varie caratteristiche dello standard, particolare attenzione andrà nella creazione di un prodotto *portabile* ed *usabile*.

- **Portabilità:** è uno dei requisiti importanti specificati dai proponenti. Il prodotto dovrà funzionare nei vari sistemi operativi<sub>G</sub> specificati nell'*Analisi dei Requisiti v4.0.0*.
- **Usabilità:** vista l'aleatorietà di questa caratteristica, è difficile trovare una misura per garantirne il soddisfacimento. Si farà riferimento quindi ad un'interazione con i proponenti, attraverso dei prototipi di interfaccia grafica per avere al più presto dei *feedback* a riguardo.

## 2.2 Organizzazione

L'attività di verifica, necessaria al fine di garantire la qualità di un processo o prodotto, verrà istanziata ogni qualvolta il prodotto di un processo raggiungerà uno stato ritenuto diverso da quello precedente. La verifica potrà essere più circoscritta e precisa possibile, poiché applicata solamente alle modifiche effettuate su una versione precedentemente approvata del prodotto. Gli eventuali difetti e/o anomalie riscontrati in un prodotto, verranno trattati dal processo di risoluzione dei problemi. Il processo di verifica, diverso nelle varie fasi del progetto (vedi *Piano di Progetto v4.0.0*), sarà rispettivamente il seguente:

- **A (FA):** in tale fase si dovranno seguire i metodi di verifica descritti nelle *Norme di Progetto v4.0.0* sui documenti prodotti. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.1;

---

<sup>2</sup>Per una descrizione dettagliata dello standard, consultare l'appendice A.2

- **B (FB):** in questa fase si andranno a verificare i processi che hanno portato ad un eventuale incremento nel versionamento dei documenti redatti nella precedente fase, rispettando le procedure descritte nelle *Norme di Progetto v4.0.0*. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.2;
- **C (FC):** in questa fase, oltre a verificare i processi che hanno portato ad un eventuale incremento nel versionamento dei documenti, si andranno a verificare i prodotti e i processi attuati per l'attività di progettazione dell'architettura, seguendo le procedure descritte nelle *Norme di Progetto v4.0.0*. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.3;
- **D (FD):** oltre alla verifica della documentazione, da eseguire in maniera analoga dei punti precedenti, in questa fase si verificherà che ogni requisito sia rintracciabile da uno dei componenti emersi durante la fase C. Si dovranno rispettare inoltre le norme di codifica, descritte nelle *Norme di Progetto v4.0.0*. I risultati che emergeranno da tali attività, saranno descritti nell'appendice B.4;
- **E (FE):** oltre all'usuale verifica della documentazione, in questa fase verrà effettuato il collaudo del prodotto, garantendo la correttezza dello stesso, attraverso gli strumenti descritti nelle *Norme di Progetto v4.0.0*. Gli esiti di tali attività verranno descritti nell'appendice E.

## 2.3 Pianificazione strategica e temporale

Un'attività di verifica ben organizzata e sistematica, può garantire un alto livello di qualità del prodotto e di minimizzare il rischio di non rispettare le scadenze fissate nel *Piano di Progetto v4.0.0*. Al fine di semplificare l'attività di verifica, è opportuno che prima di iniziare la redazione di un documento o l'attività di codifica, siano ben chiari la struttura e i contenuti degli stessi. È inoltre opportuna un'attenta lettura delle *Norme di Progetto v4.0.0*, dove saranno descritte le varie metodologie da seguire per l'individuazione e la correzione degli errori.

## 2.4 Responsabilità

Le responsabilità di tutte le attività di verifica e validazione sono a carico del *Responsabile di Progetto* e dei *Verificatori*. Durante le varie fasi di progetto, questi ruoli saranno assunti da diversi componenti del gruppo, come descritto nel *Piano di Progetto v4.0.0*.

## 2.5 Risorse necessarie

L'utilizzo di risorse **umane** e **tecnologiche** è indispensabile per l'attuazione delle verifiche. I ruoli necessari per garantire qualità e correttezza sono:

- **Responsabile di Progetto:** coordina le attività di verifica;
- **Verificatore:** esegue la verifica della documentazione ed effettua test per garantire il corretto funzionamento del software;
- **Amministratore di Progetto:** definisce le norme e le metodologie per le attività di verifica;
- **Programmatore:** effettua i test sul codice da lui prodotto e apporta eventuali modifiche approvate dall'Amministratore o dal Responsabile.

Per una descrizione dettagliata dei ruoli e delle rispettive responsabilità, fare riferimento al *Piano di Progetto v4.0.0*.

Nelle *Norme di Progetto v4.0.0* sono descritte le risorse **tecnologiche**, ossia gli strumenti hardware e software, necessari alle attività di verifica delle varie fasi. In particolare esse saranno automatizzate il più possibile, al fine di garantire un processo di verifica più semplice e corretto, congiunto ad una mole di lavoro minore per i verificatori.

## 2.6 Tecniche, misure e metriche

### 2.6.1 Tecniche di analisi

Le tecniche di analisi che verranno adottate sono due: l'*analisi statica* e l'*analisi dinamica*, entrambe descritte approfonditamente nelle *Norme di Progetto v4.0.0*.

### 2.6.2 Misure e metriche di qualità

Al fine di garantire qualità, è necessario fissare delle metriche sulle quali misurare i risultati ottenuti dalle varie attività di verifica. Verranno qui descritte quindi, le metriche che il gruppo intende adottare nel corso della progettazione e realizzazione del prodotto. Vista l'iniziale inesperienza del gruppo e visto il ciclo di vita adottato (vedo *Piano di Progetto v4.0.0*), qualora ci fossero metriche incerte ed approssimative, esse verranno migliorate in modo incrementale. Per ogni metrica verrà specificato anche il suo range di **accettazione**, ossia un intervallo entro il quale un prodotto si può ritenere soddisfacente, oltre che a un range **ottimale**, ossia un valore entro il quale dovrebbe arrivare la misurazione del prodotto.

**Metriche per i processi** Come metrica per i processi si è deciso di utilizzare indici che analizzano sia i costi che i tempi. Questi indici vengono utilizzati anche per tenere sotto controllo i processi durante il loro svolgimento; sono quindi descritti nel *Piano di Progetto v4.0.0*. Le metriche in questione sono:

- **Schedule Variance<sub>G</sub> (SV<sub>G</sub>):** indica se si è in linea, in anticipo o in ritardo rispetto alla pianificazione temporale delle attività. SV<sub>G</sub> è un mero indicatore di efficacia. Se  $> 0$  significa che il gruppo produce con maggiore velocità rispetto alla pianificazione, viceversa se negativo. Il range di accettazione è  $[>= -(\text{costo preventivo fase} \times 5\%)]$  mentre il range ottimale è  $[>= 0]$ ;
- **Budget Variance<sub>G</sub> (GV<sub>G</sub>):** indica se alla data corrente (in qui avviene la misurazione) si è speso di più o meno rispetto a quanto è stato pianificato. BV<sub>G</sub> è un indicatore che ha valore contabile e finanziario. Se  $> 0$  significa che si sta consumando il proprio budget con minore velocità rispetto a quanto pianificato, viceversa se è negativo. Il range di accettazione è  $[>= -(\text{costo preventivo fase} \times 10\%)]$  mentre il range ottimale è  $[>= 0]$ .

**Metriche per la documentazione.** Come metrica per i documenti redatti si è scelto di adottare un indice di leggibilità; l'indice scelto è quello di Gulpease, che oltre ad essere tarato specificamente sulla lingua italiana, ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. L'indice di Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere. La formula per il suo calcolo è la seguente:

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

I risultati sono compresi tra 0 e 100, dove 100 indica la leggibilità più alta e 0 la leggibilità più bassa. In generale risulta che i testi con indice

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

A fronte di questi dati si è deciso di fissare un *range* di accettazione di [40-100] e uno ottimale [50-60].

**Metriche per il software.** Verranno qui elencate le metriche che si intendono adottare per garantire qualità del software. Esse rappresentano gli obiettivi che si cercheranno di raggiungere in quanto a qualità. Il gruppo si riserva la facoltà di apportare modifiche a tali metriche nel corso delle varie revisioni, visti i motivi riguardanti l'inesperienza, già precedentemente citati.

- **Complessità ciclomatica:** misura direttamente il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso; i nodi del grafo corrispondono a gruppi indivisibili di istruzioni e gli archi connettono due nodi se le istruzioni di un nodo possono essere eseguite dopo le istruzioni dell'altro nodo; Come range di accettazione si è deciso di fissare [1-15], mentre come range ottimale [1-10].
- **Numero di parametri:** indica il numero di parametri formali di un metodo. Un metodo con un numero alto di parametri può indicare la necessità di introdurre un nuovo metodo a cui associare certe funzionalità interne. Il range di accettazione è [0-8], mentre quello ottimale è [0-4];
- **Numero di attributi di una classe:** indica il numero di attributi interni di una classe. Un numero troppo elevato potrebbe indicare la necessità di suddividere la classe in una gerarchia. Come range di accettazione si è deciso di fissare [0-15], mentre come range ottimale [3-9];
- **Linee di codice per linee di commento:** un codice poco commentato comporta una difficile manutenibilità; questa metrica indica il rapporto tra il numero di linee di codice per il numero di linee di commento. Il range di accettazione è [ $>0.20$ ], quello ottimale è [ $>0.30$ ];
- **Logical SLOC:** misura le dimensioni del software basandosi sul numero di linee di codice sorgente. Ci sono due tipi di SLOC, quello *physical* e quello *logical*: si userà il secondo, che conta solamente gli *statement*, ignorando le righe vuote, quelle bianche e i commenti; il range di accettazione è [1-50], mentre quello ottimale è [1-25];
- **Numero di livelli di annidamento:** rappresenta il numero di livelli di annidamento dei metodi. Un numero elevato riduce il livello di astrazione del codice e comporta un'elevata complessità dello stesso. Il range di accettazione è [1-6], mentre quello ottimale è [1-3];
- **Accoppiamento**
  - **Accoppiamento afferente:** viene calcolato per ogni package<sub>G</sub> dell'architettura. Indica il numero di package<sub>G</sub> contenente classi, che hanno delle dipendenze con delle classi interne al package<sub>G</sub> oggetto dell'analisi. Il valore di tale indice è direttamente proporzionale al grado di dipendenza del resto del software dal package<sub>G</sub> preso in considerazione.  
Tale indice dovrebbe avere un valore ragionevole. Se risulta troppo basso, può indicare che il package<sub>G</sub> non offre sufficienti funzionalità al sistema e quindi può essere scarsamente utile. Se invece risulta troppo alto, potrebbe accadere che eventuali modifiche al package<sub>G</sub>, comportino costi elevati di adattamento delle classi che vi dipendono qualora non fosse stato progettato adeguatamente il sistema di interfacce;
  - **Accoppiamento efferente:** viene calcolato per ogni package<sub>G</sub> dell'architettura. Indica il numero di package contenente classi dalle quali dipendono le classi interne al package<sub>G</sub> in analisi. Il valore di tale indice è direttamente proporzionale al grado di indipendenza del package<sub>G</sub> preso in considerazione.  
In generale, tale indice andrebbe tenuto basso. L'obiettivo è quello di aumentare le funzionalità del package<sub>G</sub> senza duplicare i servizi che magari sono offerti da altri package<sub>G</sub>.
- **Copertura del codice:** indica la percentuale di istruzioni che saranno eseguite durante i test. Un'alta percentuale di istruzioni coperte dai test eseguiti, porta a una maggiore probabilità di avere componenti testate con una ridotta quantità di errori.

Il valore di tale indice, può essere ridotto a causa della presenza di metodi molto semplici che non richiedono di essere testati; esempi di questi metodi sono i *getter* e i *setter*. Il range di accettazione è [65%-100%] e il range ottimale è [80%-100%].

### Riassunto della quantificazione degli obiettivi

La tabella seguente riassume la quantificazione degli obiettivi riguardante le metriche individuate.

Metrica	Tipologia	Range Accettazione	Range Ottimale
Schedule Variance	processi	[>=-(5% costo prev. fase)]	[>=0]
Budget Variance	processi	[>=-(10% costo prev. fase)]	[>=0]
Indice Gulpease	documenti	[40-100]	[50-60]
Complessità Ciclomantica	software	[1-15]	[1-10]
Numero di Parametri	software	[0-8]	[0-4]
Attributi della Classe	software	[0-15]	[3-9]
Linee Codice per Linee Commento	software	[>0.20]	[>0.30]
Logical SLOC	software	[1-50]	[1-25]
Livelli di Annidamento	software	[1-6]	[1-3]
Copertura del Codice	software	[65%-100%]	[80%-100%]

**Tabella 3:** Quantificazione degli obiettivi

## A Standard di qualità

### A.1 Standard ISO/IEC 15504

Lo standard ISO<sub>G</sub>/ IEC<sub>G</sub>15504, definisce un modello per la valutazione della *maturità* dei processi, indicando una metrica per misurarla. La scala di misura fornita dallo standard, si basa sul principio che un'alta *capacità di processo*<sub>G</sub> è associata ad un miglioramento prestazionale dello stesso.

Per ogni processo, si definisce un livello di capacità, basato sulla seguente scala a sei livelli, ognuno con i propri attributi:

- **Livello 0 - Incomplete process:** il processo non raggiunge i risultati aspettati; non ci sono prodotti identificabili come risultati del processo;
- **Livello 1 - Performed process:** lo scopo del processo viene parzialmente raggiunto. Il risultato però, potrebbe non essere stato rigorosamente pianificato e tracciato; si riescono tuttavia ad identificare i prodotti del processo
  - *Process Performance:* capacità di un processo di raggiungere gli obiettivi e di rendere identificabili i suoi risultati.
- **Livello 2 - Managed process:** il processo produce prodotti di qualità, rispettando i tempi stabiliti. I prodotti sono conformi agli standard ed ai requisiti. La principale differenza con il livello precedente, consiste nella pianificazione delle prestazioni del prodotto attraverso processi definiti
  - *Performance Management:* misura la capacità del processo di produrre prodotti entro il tempo prestabilito e le risorse disponibili;
  - *Work Product Management:* misura la capacità del processo di produrre prodotti documentati, controllati e verificati.
- **Livello 3 - Established process:** il processo viene eseguito rispettando i principi dell'ingegneria del software. La principale differenza con il livello precedente consiste nell'adozione di un *processo standard* per pianificare e gestire i processi
  - *Process Definition:* è la misura in cui l'esecuzione del processo aderisce al processo standard prefissato;
  - *Process Deployment:* è la misura in cui l'esecuzione del processo usa efficacemente adeguate risorse umane e tecniche.
- **Livello 4 - Predictable process:** il processo viene eseguito costantemente entro definiti limiti di controllo e vengono misurate dettagliatamente le prestazioni dello stesso; la qualità dei prodotti del processo è quantitativamente nota
  - *Process Measurement:* le misure ricavate dal processo vengono utilizzate per garantire il raggiungimento degli obiettivi stabiliti;
  - *Process Control:* il processo viene controllato per effettuare eventuali correzioni e miglioramenti.
- **Livello 5 - Optimizing process:** le prestazioni del processo sono ottimali e raggiunge la ripetibilità nel conseguimento degli suoi obiettivi
  - *Process Change:* l'introduzione di cambiamenti deve minimizzare il rischio di eventuali peggioramenti nel processo;
  - *Continuous Improvement:* si adotta un approccio proattivo nell'identificare eventuali modifiche per migliorare i processi esistenti.

Ogni attributo è misurabile e lo standard fissa la seguente scala di valutazione:

- **Not achieved** (0-15%);
- **Partially achieved** (>15%-50%);



- **Largely achieved** (>50%-85%);
- **Fully achieved** (85%-100%).

## A.2 Standard ISO/IEC 9126

Lo standard ISO<sub>G</sub>/IEC<sub>G</sub>9126 delinea una serie di normative atte a descrivere un modello di qualità del software. Il modello propone un approccio alla qualità focalizzato a favorire un miglioramento dell'organizzazione e dei processi e, come conseguenza concreta, della qualità del prodotto sviluppato.

Il modello di qualità stabilito dallo standard è classificato in base a sei caratteristiche generali:

1. **Funzionalità:** è la capacità del prodotto software di fornire soluzioni che soddisfino esigenze stabilite
  - **Appropriatezza:** il prodotto software fornisce un appropriato insieme di funzioni per i compiti specificati e gli obiettivi prefissati;
  - **Accuratezza:** il prodotto software fornisce i risultati richiesti;
  - **Interoperabilità:** il prodotto software interagisce ed opera con uno o più sistemi specificati;
  - **Conformità:** il prodotto software aderisce a standard, convenzioni e regolamentazioni rilevanti al settore operativo a cui vengono applicate;
  - **Sicurezza:** il prodotto software protegge le informazioni e i dati contenuti, negando che persone o sistemi non autorizzati possano accedervi o modificarli e li mette a disposizione dei soli autorizzati.
2. **Affidabilità:** è la capacità del prodotto software di mantenere uno specificato livello di prestazioni
  - **Maturità:** non si verificano errori e malfunzionamenti durante l'uso del prodotto;
  - **Tolleranza agli errori:** il prodotto riesce a mantenere livelli prefissati di prestazioni anche in presenza di eventuali malfunzionamenti;
  - **Recuperabilità:** il prodotto riesce a recuperare delle informazioni rilevanti a seguito di un malfunzionamento;
  - **Aderenza:** il prodotto aderisce a standard, regole e convenzioni in ambito di affidabilità.
3. **Efficienza:** è la capacità di fornire prestazioni relativamente alla quantità di risorse usate
  - **Comportamento rispetto al tempo:** i tempi di risposta del prodotto sono ragionevoli rispetto alla richiesta effettuata;
  - **Efficienza:** l'utilizzo delle risorse è adeguato alle attività da svolgere;
  - **Conformità:** il prodotto aderisce a standard, regole e convenzioni in ambito di affidabilità.
4. **Usabilità:** è la capacità del prodotto di essere capito, appreso e usato dall'utente
  - **Comprensibilità:** esprime la facilità di comprensione delle funzionalità del prodotto;
  - **Apprendibilità:** l'uso prodotto risulta essere di facile apprendimento da parte degli utenti;
  - **Operabilità:** capacità di mettere gli utenti in condizione di fare uso delle funzionalità del prodotto;
  - **Attrattività:** il prodotto riesce ad attrarre gli utenti;

- **Conformità:** il prodotto aderisce a standard, regole e convenzioni in ambito di usabilità.

**5. Manutenibilità:** è la capacità del prodotto di essere manutenibile nel corso del tempo, al fine di consentire la correzione degli errori o il rilascio di nuove funzionalità

- **Analizzabilità:** indice di facilità con la quale è possibile analizzare il codice per localizzare eventuali errori;
- **Modificabilità:** capacità del prodotto di permettere l'implementazione di future modifiche;
- **Stabilità:** capacità di evitare effetti inaspettati derivanti da modifiche errate;
- **Testabilità:** il prodotto è facilmente testabile per validare le modifiche apportate.

**6. Portabilità:** è la capacità del software di operare in diversi ambienti di lavoro

- **Adattabilità:** il software si adatta a differenti ambienti senza dover applicare modifiche diverse da quelle fornite;
- **Installabilità:** capacità del software di essere installato in uno specifico ambiente;
- **Conformità:** il prodotto aderisce a standard, regole e convenzioni in ambito di usabilità;
- **Sostituibilità:** il prodotto riesce a sostituire i prodotti analoghi già esistenti.

## B Resoconto delle attività di verifica

Qui verranno riportati gli esiti delle attività di verifica effettuate durante le varie fasi del progetto.

### B.1 Revisione dei Requisiti

#### B.1.1 Documenti

A seguito delle attività di verifica effettuate sulla documentazione, sono state ricavate le seguenti misurazioni:

Documento	Indice Gulpease	Esito
<i>Piano di Qualifica v1.0.0</i>	49	<i>Superato</i>
<i>Piano di Progetto v1.0.0</i>	48	<i>Superato</i>
<i>Analisi dei Requisiti v1.0.0</i>	56	<i>Superato</i>
<i>Norme di Progetto v1.0.0</i>	54	<i>Superato</i>
<i>Studio di Fattibilità v1.0.0</i>	52	<i>Superato</i>
<i>Glossario v1.0.0</i>	55	<i>Superato</i>

**Tabella 4:** Indice Gulpease dei documenti presentati in Revisione dei Requisiti

Tutti i documenti verificati, soddisfano i range di accettazione specificati nella sezione 2.6.2. Alcuni di essi però, non rientrano nel range ottimale. Questo implica quindi, una futura ricerca e modifica di frasi troppo complesse, al fine di migliorare la lettura dei documenti e rientrare nel range ottimale.

## B.2 Revisione di Progettazione

### B.2.1 Documenti

A seguito delle attività di verifica effettuate sulla documentazione, sono state ricavate le seguenti misurazioni:

Documento	Indice Gulpease	Esito
<i>Piano di Qualifica v2.0.0</i>	55	<i>Superato</i>
<i>Piano di Progetto v2.0.0</i>	53	<i>Superato</i>
<i>Analisi dei Requisiti v2.0.0</i>	65	<i>Superato</i>
<i>Norme di Progetto v2.0.0</i>	57	<i>Superato</i>
<i>Specifica Tecnica v1.0.0</i>	49	<i>Superato</i>
<i>Glossario v2.0.0</i>	55	<i>Superato</i>

**Tabella 5:** Indice Gulpease dei documenti presentati in Revisione di Progettazione

Tutti i documenti verificati, soddisfano i range di accettazione specificati nella sezione 2.6.2. Alcuni di essi però, non rientrano nel range ottimale. Questo implica quindi, una futura ricerca e modifica di frasi troppo complesse, al fine di migliorare la lettura dei documenti e rientrare nel range ottimale.

## B.3 Revisione di Qualifica

### B.3.1 Documenti

A seguito delle attività di verifica effettuate sulla documentazione, sono state ricavate le seguenti misurazioni:

Documento	Indice Gulpease	Esito
<i>Piano di Qualifica v3.0.0</i>	55	<i>Superato</i>
<i>Piano di Progetto v3.0.0</i>	52	<i>Superato</i>
<i>Analisi dei Requisiti v3.0.0</i>	65	<i>Superato</i>
<i>Norme di Progetto v3.0.0</i>	55	<i>Superato</i>
<i>Specifica Tecnica v2.0.0</i>	51	<i>Superato</i>
<i>Glossario v3.0.0</i>	55	<i>Superato</i>
<i>Manuale Utente v1.0.0</i>	59	<i>Superato</i>
<i>Definizione di Prodotto v1.0.0</i>	50	<i>Superato</i>

**Tabella 6:** Indice Gulpease dei documenti presentati in Revisione di Qualifica

Tutti i documenti verificati.

### B.3.2 Progettazione

Viene di seguito riportata una tabella, nella quale è indicato il livello di accoppiamento afferente ed efferente, individuato per ogni componente rilevato durante la progettazione architettuale.

Componente	Afferente	Efferente
Romeo::Model	0	0
Romeo::Model::Core	5	2
Romeo::Model::Core::Algorithm	2	0
Romeo::Model::Core::Feature	2	0
Romeo::Model::Util	0	0
Romeo::Model::Util::Log	1	2
Romeo::Model::Util::ReaderModel	1	0
Romeo::Model::Util::ExporterModel	1	0
Romeo::Model::Util::DAO	1	3
Romeo::Model::QtModel	1	0
Romeo::Model::Help	2	0
Romeo::View	0	0
Romeo::View::Dialog	1	1
Romeo::View::Component	1	1
Romeo::View::Window	1	1
Romeo::Controller	0	2

### B.3.3 Codice

In questa sezione vengono riportati i risultati dei test di analisi statica e analisi dinamica effettuati sul codice finora prodotto. Per ogni test effettuato vengono indicati i valori medi ed i valori massimi ottenuti, giustificando qualora ci fossero valori non rientranti nel range di accettazione.

#### Complessità ciclomatica

- **Media:** 4
- **Massimo:** 10

Tutti i moduli rientrano nel range di accettazione.

#### Parametri per metodo

- **Media:** 2.5
- **Massimo:** 5

Tutti i moduli rientrano nel range di accettazione.

#### Attributi per classe

- **Media:** 6.5
- **Massimo:** 10

In questo caso alcuni moduli del package view non rientrano nel range di accettazione, questo è dovuto al fatto che le classi view sono composte da un gran numero di elementi per offrire tutte le funzionalità individuate in analisi dei requisiti.

#### Linee di codice per linee di commento

- **Media:** 1.5

Tutti i moduli rientrano nel range di accettazione.

### Logical SLOC

- **Media:** 30
- **Massimo:** 60

In questo caso alcuni moduli, soprattutto del package view, non rientrano nel range di accettazione, questo è dovuto a causa dell'elevata quantità di attributi che questi possiedono, e che vengono utilizzati per fornire le funzionalità individuate in analisi dei requisiti.

### Livello di annidamento

- **Media:** 2
- **Massimo:** 5

Tutti i moduli rientrano nel range di accettazione.

### Copertura del codice

- **Media:** 65

La percentuale di copertura del codice è stata calcolata tenendo conto del numero effettivo di linee di codice testate con i test di unità. Risultano quindi esclusi dal calcolo tutti i metodi *getter* e *setter*, oltre ai metodi virtuali puri, in quanto, data la loro semplicità, non sono stati considerati meritevoli di test.

## B.4 Revisione di Accettazione

### B.4.1 Documenti

A seguito delle attività di verifica effettuate sulla documentazione, sono state ricavate le seguenti misurazioni:

Documento	Indice Gulpease	Esito
<i>Piano di Qualifica v4.0.0</i>	55	<i>Superato</i>
<i>Piano di Progetto v4.0.0</i>	56	<i>Superato</i>
<i>Analisi dei Requisiti v3.0.0</i>	65	<i>Superato</i>
<i>Norme di Progetto v4.0.0</i>	56	<i>Superato</i>
<i>Specifica Tecnica v3.0.0</i>	54	<i>Superato</i>
<i>Glossario v4.0.0</i>	55	<i>Superato</i>
<i>Manuale Utente v2.0.0</i>	58	<i>Superato</i>
<i>Definizione di Prodotto v2.0.0</i>	51	<i>Superato</i>

**Tabella 7:** Indice Gulpease dei documenti presentati in Revisione di Accettazione

Tutti i documenti verificati.

### B.4.2 Codice

In questa sezione vengono riportati i risultati dei test di analisi statica e analisi dinamica effettuati sul codice finale. Per ogni test effettuato vengono indicati i valori medi ed i valori massimi ottenuti, giustificando qualora ci fossero valori non rientranti nel range di accettazione.

#### Complessità ciclomatica

- **Media:** 6
- **Massimo:** 9

Tutti i moduli rientrano nel range di accettazione.

#### Parametri per metodo

- **Media:**5
- **Massimo:**6

Tutti i moduli rientrano nel range di accettazione.

#### Attributi per classe

- **Media:**5
- **Massimo:**18

In questo caso alcuni moduli del package view non rientrano nel range di accettazione, questo è dovuto al fatto che le classi view sono composte da un gran numero di elementi per offrire tutte le funzionalità individuate in analisi dei requisiti e per la posizione dei componenti grafici nella gui<sub>G</sub>.

#### Linee di codice per linee di commento

- **Media:**1.76365

### Logical SLOC

- **Media:**28
- **Massimo:**60

In questo caso alcuni moduli, soprattutto del package view, non rientrano nel range di accettazione, questo è dovuto a causa dell'elevata quantità di attributi che questi possiedono, e che vengono utilizzati per fornire le funzionalità individuate in analisi dei requisiti.

### Livello di annidamento

- **Media:**2
- **Massimo:**6

### Copertura del codice

- **Media:** 87%

La percentuale di copertura del codice è stata calcolata tenendo conto del numero effettivo di linee di codice testate con i test di unità. Risultano quindi esclusi dal calcolo tutti i metodi *getter* e *setter*, oltre ai metodi virtuali puri, in quanto, data la loro semplicità, non sono stati considerati meritevoli di test.

#### B.4.3 Verifica della copertura dei requisiti

La tabella seguente riporta la copertura dei requisiti; come da *verbale 8 esterno del 04/06/2014* alcuni requisiti hanno subito variazione di importanza.

Categoria	Non Soddisfatti	Soddisfatti	Totali
Obbligatorio	3	150	153
Opzionale	1	5	6
Desiderabile	22	4	26
<b>Totali</b>	26	159	185

**Tabella 8:** Riepilogo copertura dei requisiti



## C Dettaglio dell'esito delle revisioni

A seguito di ogni revisione, il committente pubblicherà una valutazione sull'andamento del progetto e sui documenti consegnati. Il gruppo quindi dovrà procedere ad una correzione laddove questa sia necessaria, in modo tale da proseguire i lavori su una base solida e corretta. Di seguito sono elencate le modifiche effettuate a seguito delle revisioni.

### C.1 Revisione dei Requisiti

- **Norme di Progetto:**

- sono stati inseriti vari diagrammi di flusso per rendere la lettura del documento più agevole e facilitarne la comprensione;
- la struttura del documento è stata cambiata, suddividendolo per procedure, attività, processi e strumenti;
- sono state integrate le parti riguardanti le tecniche di analisi, erroneamente descritte nel *Piano di Qualifica*.

- **Piano di Progetto:**

- la qualità delle immagini è stata migliorata;
- corretto l'utilizzo della terminologia.

- **Piano di Qualifica:**

- come suggerito, le parti riguardanti le tecniche adottate, sono state spostate nelle *Norme di Progetto*;
- è stata aggiunta la sezione *Definizione Obiettivi*.

- **Analisi dei Requisiti:**

- sono state apportate le modifiche suggerite, migliorando e descrivendo i casi d'uso;
- aggiunti i requisiti di qualità;
- corretti alcuni requisiti.

### C.2 Revisione di Progettazione

- **Norme di Progetto:**

- sono stati inseriti i meccanismi di controllo e rendicontazione, erroneamente descritti nel *Piano di Progetto*;
- è stata sistemata la struttura del documento suddividendo i contenuti sulla base dei processi che si è deciso di adottare.

- **Piano di Progetto:**

- modificata la struttura del documento integrando alcuni argomenti semanticamente correlati, che prima risultavano slegati tra di loro;
- aggiornata la sessione riguardante l'analisi dei rischi;
- esplicitato il preventivo a finire;
- corretto l'utilizzo della terminologia, particolare attenzione va data al nome delle fasi che prima sviavano confondendo le fasi, con le attività in esse svolte.

- **Piano di Qualifica:**

- aggiunte le metriche riguardanti i processi;
- adattato lo standard ISO/IEC 15504 ai processi che vengono adottati.

- **Analisi dei Requisiti:**
  - incrementata sezione riguardante i diagrammi delle attività;
  - sistemati i casi d'uso non corretti segnalati dal committente;
  - sistemata tipologia del requisito segnalato dal committente.
- **Specifica Tecnica:**
  - migliorata la contestualizzazione dei design pattern<sub>G</sub>;
  - descrizione più approfondita delle relazioni tra le componenti logiche;
  - sistemati i diagrammi delle classi per farli aderire ai design pattern<sub>G</sub> associati;
  - sistemati i diagrammi delle classi evidenziando le classi astratte e le interfacce;
  - sistemata terminologia riguardante il database.

### C.3 Revisione di Qualifica

- **Norme di Progetto:**
  - inserite le tecniche di verifica in un nuovo capitolo *Processo di Verifica* erroneamente descritti nel *Piano di Qualifica*;
  - inseriti i contenuti riguardanti la gestione amministrativa della revisione erroneamente descritta nel *Piano di Qualifica*;
  - integrati i contenuti del precedente capitolo 2 all'interno dei processi organizzativi;
  - suddivisi i contenuti del precedente capitolo 6 in relazione alle attività che supportano.
- **Piano di Progetto:**
  - esplicitato il preventivo a finire.
- **Piano di Qualifica:**
  - esplicitati gli obiettivi riguardanti le metriche descritte nel capitolo *Misure e metriche di qualità*.
- **Analisi dei Requisiti:**
  - sistemata la grandezza dell'immagine come segnalato dal committente.
- **Specifica Tecnica:**
  - migliorata la contestualizzazione del design pattern<sub>G</sub> DAO;
  - sistemate le problematiche legate alla progettazione di alcune componenti.
- **Definizione di Prodotto:**
  - migliorata la contestualizzazione dei design pattern<sub>G</sub>;
  - sistemata la terminologia da Package<sub>G</sub> a Diagramma dei Package<sub>G</sub>;
  - sistemati gli errori commessi nella rappresentazione dei diagrammi delle classi non conforme allo standard UML 2.0;
  - sistemati i diagrammi di sequenza in modo da favorire maggiormente la lettura;
  - dettagliata maggiormente la descrizione dei metodi segnalati dal committente in sede di revisione di qualifica;
  - sistemate incongruenze tra la descrizione dei metodi e la loro rappresentazione nei diagrammi delle classi.
- **Manuale Utente:**
  - è stato inserito la procedura di installazione del prodotto;
  - è stata descritta con maggior dettaglio l'esecuzione di un'analisi come segnalato in sede di revisione di qualifica.

## D Pianificazione dei test

Di seguito verranno descritti tutti i test di integrazione, sistema e validazione che il gruppo *Seven Monkeys* intende effettuare. È previsto un futuro aggiornamento per quanto riguarda i test di unità. Nel *Piano di Progetto v4.0.0* vengono specificate le tempistiche di esecuzione dei test.

Nelle tabelle sottostanti la colonna dello *Stato* dei test riporterà il valore **N.I.** per i test che devono ancora essere effettuati; essi verranno eseguiti successivamente, come descritto nel *Piano di Progetto v4.0.0*.

### D.1 Test di sistema

In questa sezione vengono descritti i test di sistema, il cui obbiettivo è quello di verificare il comportamento dinamico dell'intero sistema, rispetto ai requisiti descritti nel documento *Analisi dei Requisiti v4.0.0*. Vengono qui di seguito riportati i test di sistema ritenuti meritevoli di un test, al fine di garantire il soddisfacimento dei requisiti software individuati.

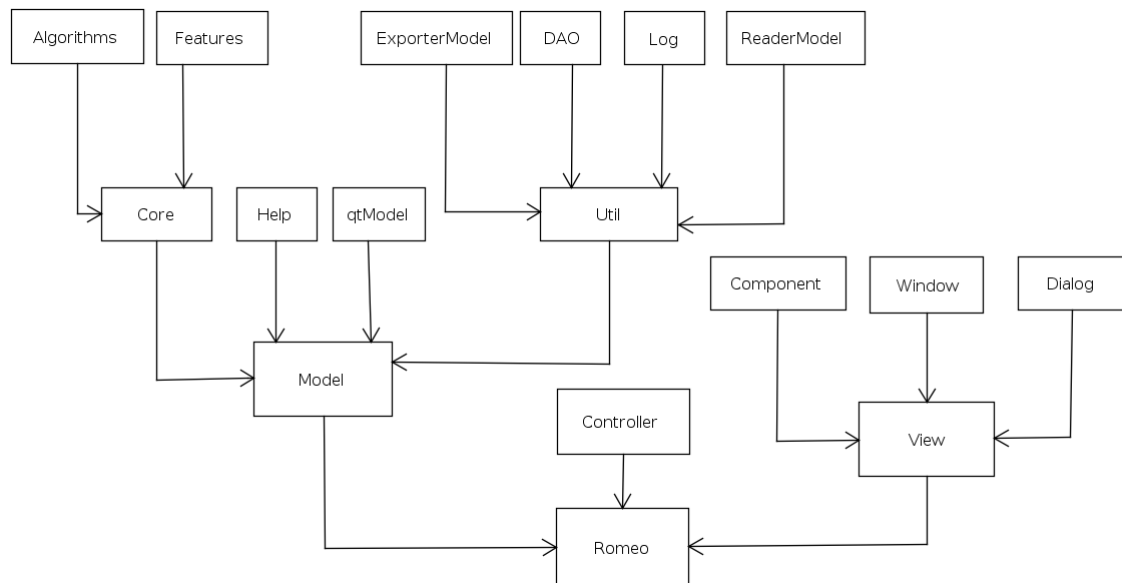
### D.1.1 Descrizione test di sistema

Test	Descrizione	Stato	Requisito
TS1.2.1.1	Viene verificato che il sistema accetti in input file di formato PNG	superato	R0F1.2.1.1
TS1.2.1.2	Viene verificato che il sistema accetti in input file di formato JPG	superato	R0F1.2.1.2
TS1.2.1.3	Viene verificato che il sistema accetti in input file di formato BMP	superato	R0F1.2.1.3
TS1.2.1.4	Viene verificato che il sistema accetti in input file di formato AVI	superato	R0F1.2.1.4
TS1.2.1.5	Viene verificato che il sistema accetti in input file di formato NIfTI	superato	R0F1.2.1.5
TS1.2.1.6	Viene verificato che il sistema accetti in input file di formato Analyze7.5	superato	R0F1.2.1.6
TS1.3.1	Viene verificato che il sistema accetti in input file di formato PNG per le maschere	superato	R0F1.3.1
TS1.3.2	Viene verificato che il sistema accetti in input file di formato JPG per le maschere	superato	R0F1.3.2
TS1.3.3	Viene verificato che il sistema accetti in input file di formato BMP per le maschere	superato	R0F1.3.3
TS1.3.4	Viene verificato che il sistema accetti in input file di formato NIfTI per le maschere	superato	R0F1.3.4
TS1.3.5	Viene verificato che il sistema accetti in input file di formato Analyze7.5 per le maschere	superato	R0F1.3.5
TS1.4	Viene verificato che il sistema visualizzi correttamente un errore quando si cerca di caricare un'immagine di formato non consentito	N.I.	R0F1.4
TS10.2.1	Viene verificato che il software, se richiesto, mostri il risultato delle features appena processate	superato	R0F10.2.1
TS12.2	Viene verificato che i risultati delle analisi effettuate vengano esportati nel formato corretto	superato	R0F12.2
TS13.1	Viene verificato che vengano visualizzate correttamente le immagini 2D all'interno del software	superato	R0F13.1
TS13.2	Viene verificato che vengano visualizzate correttamente le immagini 3D all'interno del software	superato	R0F13.2
TS16	Viene verificato che il sistema funzioni su Windows 7 32bit senza service pack o superiore	superato	R0V16
TS17	Viene verificato che il sistema funzioni su Windows 7 64bit senza service pack o superiore	superato	R0V17
TS18	Viene verificato che il sistema funzioni su Ubuntu 12.04 32bit o superiore	superato	R0V18
TS19	Viene verificato che il sistema funzioni su Ubuntu 12.04 64bit o superiore	superato	R0V19
TS20	Viene verificato che il sistema funzioni su Mac OS X 10.9 o superiore	N.I.	R0V20
TS21	Viene verificato che i risultati vengano visualizzati senza perdita di qualità	superato	R0V21

**Tabella 9:** Tracciamento test di sistema-requisiti

## D.2 Test di integrazione

In questa sezione vengono descritti i test di integrazione, necessari per verificare che il comportamento di più unità combinate sia corretto. L'approccio utilizzato è di tipo *bottom-up*, che consiste nell'integrare prima le unità a livello più basso, ossia quelle che rappresentano la logica di base del sistema e in seguito quelle ai livelli superiori. Le unità base, conosciute anche come *moduli di unità*, si riferiscono ai requisiti obbligatori e quindi alle funzionalità più importanti del sistema. Adottando questo tipo di approccio, è quindi possibile verificare più volte il comportamento di queste unità, massimizzando la ricerca di eventuali errori presenti. Il diagramma seguente non rispetta il formalismo UML ma specifica graficamente la strategia d'integrazione adottata.



**Figura 1:** Diagramma informale della strategia di integrazione

Test	Descrizione	Componente	Stato
TI-Algorithms	<b>Algoritmi di clustering:</b> verificare la presenza di tutti gli algoritmi di clustering definiti nei requisiti	Algorithms	Superato
TI-Component	<b>Componenti grafici delle finestre:</b> verifica la corretta visualizzazione e responsività dei menù nelle finestre del sistema	Component	Superato
TI-Core	<b>Core del sistema:</b> verifica il corretto funzionamento delle operazioni di analisi	Core	Superato
TI-DAO	<b>Interfaccia Database:</b> verifica la corretta interazione del sistema con il database	DAO	Superato
TI-Dialog	<b>Sistema di dialogo con l'utente:</b> verifica il corretto funzionamento delle finestre di dialogo con l'utente	Dialog	Superato
TI-ExporterModel	<b>Sistema di esportazione:</b> verifica che le immagini vengano esportate correttamente	ExporterModel	Superato
TI-Features	<b>Feature Extractor:</b> verificare la presenza di tutte le feature extractor definite nei requisiti	Features	Superato
TI-Help	<b>Sistema d'aiuto:</b> verifica il corretto funzionamento delle operazioni del sistema di aiuto	Help	Superato
TI-Log	<b>Sistema di log:</b> verifica che venga creato un file di testo che riporta tutte le operazioni compiute dal sistema	Log	Superato
TI-Model	<b>Logica di Business:</b> viene verificato il funzionamento della logica di business del sistema	Model	Superato
TI-QtModel	<b>Visualizzazione dati:</b> verificare che l'interfaccia fornisca tutti e soli i dati richiesti al database	QtModel	Superato
TI-ReaderModel	<b>Sistema di importazione:</b> verifica che le immagini vengano importate correttamente.	ReaderModel	Superato
TI-Romeo	<b>Test d'integrazione finale:</b> viene testata l'integrazione di Model View e Controller	Romeo	Superato
TI-View	<b>Interfaccia grafica:</b> verifica la corretta visualizzazione dell'interfaccia grafica nella sua completezza	View	Superato
TI-Window	<b>Finestre del sistema:</b> verifica la corretta visualizzazione delle informazioni all'interno delle finestre	Window	Superato

**Tabella 10:** Test di integrazione

Componente	Test
Romeo	TI-Romeo
Romeo::Controllor	Architettura del sistema
Romeo::Model	TI-Model
Romeo::Model::Core	TI-Core
Romeo::Model::Core::Algorithms	TI-Algorithms
Romeo::Model::Core::Features	TI-Features
Romeo::Model::Help	TI-Help
Romeo::Model::QtModel	TI-QtModel
Romeo::Model::Util	Architettura del sistema
Romeo::Model::Util::DAO	TI-DAO
Romeo::Model::Util::ExporterModel	TI-ExporterModel

Romeo::Model::Util::Log	TI-Log
Romeo::Model::Util::ReaderModel	TI-ReaderModel
Romeo::View	TI-View
Romeo::View::Component	TI-Component
Romeo::View::Dialog	TI-Dialog
Romeo::View::Window	TI-Window

**Tabella 11:** Tracciamento componente-test di integrazione

### D.3 Test di validazione

In questa sezione vengono descritti i test di validazione, necessari per accertarsi che il prodotto realizzato sia conforme alle attese. Per ogni test vengono descritti i passi che un utente deve eseguire per poter testare i requisiti ad esso associati, mentre il tracciamento tra test di validazione e requisiti è riportato nel documento *Analisi dei Requisiti v4.0.0*.

#### D.3.1 Test TV1

L'utente vuole testare la possibilità di creare un nuovo Subject. All'utente è richiesto di:

- Dare un nome univoco al Subject (TV1.1)
- Caricare un'immagine o un video di formato consentito dal filesystem (TV1.2)
- Caricare un'immagine maschera di formato consentito dal filesystem (TV1.3)

#### D.3.2 Test TV2

L'utente vuole testare la possibilità di creare un nuovo gruppo di Subject. All'utente viene richiesto di:

- Dare un nome univoco al gruppo di Subject (TV2.1)
- Scegliere i Subject da inserire nel gruppo di Subject (TV2.2)

#### D.3.3 Test TV3

L'utente vuole testare la possibilità di eliminare gruppo di Subject. All'utente viene richiesto di:

- Scegliere un gruppo di Subject da eliminare (TV3.1)
- Scegliere più gruppi di Subject da eliminare (TV3.2)
- Scegliere di esportare i risultati del Gruppo di Subject prima che venga eliminato (TV3.3)

#### D.3.4 Test TV4

L'utente vuole testare la possibilità di creare un nuovo Protocol. All'utente viene richiesto di:

- Dare un nome univoco al Protocol (TV4.1)
- Scegliere una o più feature extractor (TV4.2)
  - Inserire i parametri per le feature (TV4.2.1)
- Controllare la possibilità di inserire due feature extractor uguali con parametri diversi (TV4.3)
- Scegliere uno e un solo algoritmo di clustering (TV4.4)
  - Inserire i parametri per l'algoritmo di clustering (TV4.4.1)
- Scegliere di salvare il Protocol (TV4.5)

#### D.3.5 Test TV5

L'utente vuole testare la possibilità di eliminare un Protocol. All'utente viene richiesto di:

- Scegliere uno o più Protocol ed eliminarli (TV5.1)



### D.3.6 Test TV6

L'utente vuole testare la possibilità di creare un Dataset. All'utente viene richiesto di:

- Dare un nome univoco al Dataset (TV6.1)
- Scegliere uno o più Protocol (TV6.2)
- Scegliere un gruppo di Subject (TV6.3)

### D.3.7 Test TV7

L'utente vuole testare la possibilità di svolgere un'analisi relativa ad un Dataset. All'utente viene richiesto di:

- avviare l'analisi relativa ad un Dataset (TV7.1)
  - Scegliere di visualizzare i risultati durante l'analisi (TV7.1.1)
  - Controllare che vengano mostrate le immagini appena pronte (TV7.1.2)
  - Controllare che vengano analizzati un Subject alla volta in maniera sequenziale (TV7.1.3)
  - Controllare che prima vengano applicati i feature extractor e poi gli algoritmi di clustering (TV7.1.4)
- Verificare la presenza ed il corretto funzionamento della barra di avanzamento (TV7.2)
- Verificare che vengano visualizzati i risultati al termine dell'analisi (TV7.3)

### D.3.8 Test TV8

L'utente vuole testare la possibilità di esportare i risultati delle analisi effettuate. All'utente viene richiesto di:

- Scegliere il percorso del filesystem in cui salvare i risultati relativi ai gruppi di Subject (TV8.1)
- Scegliere di voler esportare anche i risultati relativi all'applicazione dei feature extractor (TV8.2)
- Controllare nel filesystem che l'esportazione sia avvenuta con successo (TV8.3)

### D.3.9 Test TV9

L'utente vuole testare la corretta visualizzazione dei risultati all'interno del software. All'utente viene richiesto di:

- Scegliere un immagine 2D dal menù “visualizzazione” del programma (TV9.1)
- Controllare che l'immagine 3D venga visualizzata in maniera corretta (TV9.2)
- Scegliere un immagine 3D dal menù “visualizzazione” del programma (TV9.3)
- Controllare che l'immagine 3D venga visualizzata in maniera corretta (TV9.4)

### D.3.10 Test TV10

L'utente vuole testare la guida. All'utente viene richiesto di:

- Accedere alla sezione help (TV10.1)
- Consultare i riferimenti testuali verificando che i link siano funzionanti (TV10.2)

### D.3.11 Test TV11

L'utente vuole testare la possibilità di modificare gruppi di Subject. All'utente viene richiesto di:

- Scegliere il gruppo di Subject da modificare (TV11.1)
- Scegliere i subject da modificare (TV11.2)
- Verificare che i subject siano stati eliminati correttamente (TV11.3)
  - Scegliere di visualizzare il contenuto del gruppo di subject dall'apposito menù (TC11.3.1)

### D.3.12 Test TV12

L'utente vuole testare la possibilità di eliminare i Dataset. All'utente viene richiesto di:

- Scegliere i Dataset da eliminare (TV12.1)
- Eliminare i Dataset selezionati (TV12.2)
- Verificare che i Dataset siano stati eliminati correttamente dalla lista dei Dataset (TV12.3)

## D.4 Test di unità

Di seguito vengono riportati i test di unità previsti.

Test	Descrizione	Stato	Metodi
TU1	Si verifica che,passato il path di un immagine 2D di test valida,essa venga importata correttamente.Si verifica inoltre,che dato un immagine di test 2D di formato non valido,l'importazione fallisca	Superato	ImageReader() readFile2D()
TU2	Si verifica che,passato il path di un immagine 3D di test valida,essa venga importata correttamente.Si verifica inoltre,che dato un immagine di test 3D di formato non valido,l'importazione fallisca	Superato	ImageReader() readFile3D()
TU3	Si verifica che,passato il path di un video di test valido,esso venga importato correttamente.Si verifica inoltre,che dato un video di test di formato non valido,l'importazione fallisca	Superato	VideoReader() readFile2D() readFile3D()
TU4	Si verifica che,passato un path e un immagine 2D valida,essa venga esportata in maniera corretta e nella cartella indicata	Superato	ImageExporter() exportFile()
TU5	Si verifica che,passato un path e un immagine 3D valida,essa venga esportata in maniera corretta e nella cartella indicata	Superato	AnalyzeExporter() exportFile() NiftiExporter() exportFile()
TU6	Si verifica che dato un Subject ben formato,esso venga correttamente salvato nel database.Si verifica inoltre che lo stesso venga letto dal database correttamente,ottenendo un oggetto identico a quello di partenza	Superato	SubjectDAO() existSubjectlWithName() getSubjectByName() getAllSubject() getAllSubjectName() createSubject() subjectsOfGroup() getGroupOfSubject() getSubjectsByType()
TU7	Si verifica che dato un GroupOfSubject ben formato,esso venga correttamente salvato nel database.Si verifica inoltre che lo stesso venga letto dal database correttamente,ottenendo un oggetto identico a quello di partenza	Superato	GroupDAO() existGroupWithName() getAllGroup() getGroupByName() createGroup() deleteGroup() addSubjectToGroup() removeSubjectFromGroup() getSubjectOfGroup()

TU8	Si verifica che dato un Dataset ben formato,esso venga correttamente salvato nel database.Si verifica inoltre che lo stesso venga letto dal database correttamente,ottenendo un oggetto identico a quello di partenza	Superato	DatasetDAO() existSubjectWithName() getAllDataset() getAllDatasetName() getGroupOfDataset() getProtocolOfDataset() createDataset() deleteDataset() addProtocol()
TU9	Si verifica che dato un Protocol ben formato,esso venga correttamente salvato nel database.Si verifica inoltre che lo stesso venga letto dal database correttamente,ottenendo un oggetto identico a quello di partenza	Superato	ProtocolDAO() existProtocolWithName() getAllProtocol() getProtocolByName() createProtocol() deleteProtocol() getAllProtocolName() getAllProtocolOfDataset()
TU10	Si verifica che dato un Algorithm ben formato,esso venga correttamente salvato nel database.Si verifica inoltre che lo stesso venga letto dal database correttamente,ottenendo un oggetto identico a quello di partenza	Superato	AlgorithmDAO() getAlgorithmById() addAlgorithm() deleteAlgorithm() getAllAlgorithm() getAlgorithmOfProtocol()
TU11	Si verifica che dato un Analysis ben formato,esso venga correttamente salvato nel database.Si verifica inoltre che lo stesso venga letto dal database correttamente,ottenendo un oggetto identico a quello di partenza	Superato	AnalysisDAO() createAnalysis() deleteAnalysis() getAllAnalysis() getAnalysisByDate() getAnalysisOfDataset()
TU12	Si verifica che,data un InternalImage2D venga applicata correttamente,la feature extractor Standard Deviation.Per fare ciò viene confrontato il risultato,con l'immagine attesa,fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() StandardDeviationFeature() singleChannelExecution2D()
TU13	Si verifica che,data un InternalImage3D venga applicata correttamente,la feature extractor Standard Deviation.Per fare ciò viene confrontato il risultato,con l'immagine attesa,fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() StandardDeviationFeature() singleChannelExecution3D()
TU14	Si verifica che,data un InternalImage2D venga applicata correttamente,la feature extractor Skewness.Per fare ciò viene confrontato il risultato,con l'immagine attesa,fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() SkewnessFeature() singleChannelExecution2D()

TU15	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Skewness. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() SkewnessFeature() singleChannelExecution3D()
TU16	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Kurtosis. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() KurtosisFeature() singleChannelExecution2D()
TU17	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Kurtosis. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() KurtosisFeature() singleChannelExecution3D()
TU18	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Mean. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() MeanFeature() singleChannelExecution2D()
TU19	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Mean. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FirstOrderFeature getWindowSize() getParameters() getType() setParameters() MeanFeature() singleChannelExecution3D()
TU20	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Correlation. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() CorrelationFeature() singleChannelExecution2D()
TU21	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Correlation. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() CorrelationFeature() singleChannelExecution3D()

TU22	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Energy. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() EnergyFeature() singleChannelExecution2D()
TU23	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Energy. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() EnergyFeature() singleChannelExecution3D()
TU24	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Contrast. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() ContrastFeature() singleChannelExecution2D()
TU25	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Contrast. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() ContrastFeature() singleChannelExecution3D()
TU26	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Homogeneity. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() HomogeneityFeature() singleChannelExecution2D()
TU27	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Homogeneity. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() HomogeneityFeature() singleChannelExecution3D()
TU28	Si verifica che, data un InternalImage2D venga applicata correttamente, la feature extractor Entropy. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() EntropyFeature() singleChannelExecution2D()

TU29	Si verifica che, data un InternalImage3D venga applicata correttamente, la feature extractor Entropy. Per fare ciò viene confrontato il risultato, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	SecondOrderFeature() getWindowSize() getParameters() getType() setParameters() EntropyFeature() singleChannelExecution3D()
TU30	Si verifica che, data un InternalImage2D venga applicato correttamente, l'algoritmo di clustering Fuzzy C Mean. Per fare ciò viene confrontato il risultato ottenuto, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FuzzyCMeansAlgorithm() singleChannelExecution2D() getNumberOfCluster() getFuzzyIndex() getMaxIteration() getThreshold()
TU31	Si verifica che, data un InternalImage3D venga applicato correttamente, l'algoritmo di clustering Fuzzy C Mean. Per fare ciò viene confrontato il risultato ottenuto, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	FuzzyCMeansAlgorithm() singleChannelExecution3D() getNumberOfCluster() getFuzzyIndex() getMaxIteration() getThreshold()
TU32	Si verifica che, data un InternalImage2D venga applicato correttamente, l'algoritmo di clustering Hierarchical. Per fare ciò viene confrontato il risultato ottenuto, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	HierarchicalAlgorithm() singleChannelExecution2D() getDistance() getLinkageCriteria()
TU33	Si verifica che, data un InternalImage3D venga applicato correttamente, l'algoritmo di clustering Hierarchical. Per fare ciò viene confrontato il risultato ottenuto, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	HierarchicalAlgorithm() singleChannelExecution3D() getDistance() getLinkageCriteria()
TU34	Si verifica che, data un InternalImage2D venga applicato correttamente, l'algoritmo di clustering KMeans. Per fare ciò viene confrontato il risultato ottenuto, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	KMeansAlgorithm() singleChannelExecution2D() getNumberOfCluster() getNumberOfReplicates() getMaxIteration() getDistance()
TU35	Si verifica che, data un InternalImage3D venga applicato correttamente, l'algoritmo di clustering KMeans. Per fare ciò viene confrontato il risultato ottenuto, con l'immagine attesa, fornita direttamente dal proponente stesso.	Superato	KMeansAlgorithm() singleChannelExecution3D() getNumberOfCluster() getNumberOfReplicates() getMaxIteration() getDistance()
TU36	Si verifica che, dato un percorso del filesystem, e delle stringhe di prova, venga creato un file di log. In tale file devono essere stampate le stringhe attese rispetto a quelle date in input	Superato	Log() countLines() logWriter() setEnabled() writeLog()

TU37	Si verifica che data una Feature ben formata con dei parametri corretti, essa venga correttamente salvata nel database. Si verifica inoltre che la stessa venga letta dal database correttamente, ottenendo un oggetto identico a quello di partenza	Superato	FeatureDAO() createFeature() removeFeature() getFeatureOfProtocol() getAllFeatureOfProtocol()
TU38	Get-Set	Superato	setToolBar() getTypeRomeo() selectAll() setSubjectName() setImagePath() setMaskPath() setTypeSelection() getSubjectName() getMaskPath() getTypeSelected() getTableView getType() getSelectedSubjects() setNSubjects getTypeSelected() getAlgorithmSelected() getProtocolName() getFeaturesTable() getGroupTable() getProtocolsTable() getGroupInfoLabel() getProtocolInfoLabel() setProtocolInfo() setGroupInfo() getTableView() setSubjectImg() setSubjectMask() setSubjectInfo() getGroup() getSubjectsTable() getAlgorithmLabel() getFeatureLabel() setFeature() getFeature() getControllerManger() setEditFields() getMainWindowController()



TU39	Architettura di sistema	Superato	loadCss() setupMainWindow() addCentralWidget() loadCssStatic() setLayout() loadCss() setupObjectName() setupToolTip() addConnect() setupView() setupLeftFrame() setupRightFrame() loadCss() setupToolTip() addConnect() setupView() setLayout() createTop() createButtom() loadCss() setupObjectName() setupToolTip() addConnect() setupView() setupTopLayout() createTop() createButtom() setLayout() loadCss() setupObjectName() setupToolTip() addConnect() setupView() createTop() createButtom() loadCss() setupObjectName() setupToolTip() addConnect() setupView() createCenter() setupTopLayout() setLayout() setLayout() createTop() createCenter() createButtom() setupTopLayout() setupCenterLayout() loadCss() setupObjectName() setupToolTip() addConnect()
------	-------------------------	----------	--

		<div>setupView() setupLayout() createTop() createButtom() setupTable() setupInfoBlock() loadCss() setupToolTip() addConnect() setupView() itemSelected() setupLayout() createTop() createButtom() setupTopLayout() loadCss() setupObjectName() setupToolTip() addConnect() setupView() itemSelected() setupLayout() createTop() createButtom() setupTopLayout() loadCss() setupObjectName() setupToolTip() addConnect() setupView() createButtom() addConnect() setupLayout() setupTopLayout() loadCss() setupObjectName() setupToolTip() setupView() setupLayout() createTop() createButtom() setupTopLayout() loadCss() setupObjectName() setupToolTip() addConnect() setupView() setupLayout() createTop() createButtom() setupTopLayout() loadCss() setupObjectName() setupToolTip() addConnect()</div>
--	--	---

			setupView() setupLayout() loadCss() setupObjectName() setupToolTip() addConnect() setupView() createTop() createButtom() loadCss() setupObjectName() setupToolTip() addConnect() createTop() createButtom()
TU40	Signal-Slot	Superato	slotOk() slotHelp() slotBack() slotChangeSubjectName() slotChangeSubjectImage() slotChangeSubjectMask() slotSaveSubject() slotAddImage() slotAddMask() slotChangeType() slotSelectAll() slotDeselectAll() slotTypeSelection() slotSaveGroup() slotBack() slotAddFeatureClicked() slotFeatureSelected() slotOk() slotTypeChanged() slotGroupSelected() slotProtocolSelected() sloSaveDataset() slotNewSubject() slotShowSubjects() slotNewGroup() slotShowGroups() slotNewProtocol() slotShowProtocols() slotNewDataset() slotShowDatasets() slotAboutRomeo() slotShowHelp() slotAnalysisResults() slotStartAnalysis() slotWelcome() slotItemSelected() slotItemSelected() slotDeleteGroup() slotEditGroup() slotProtocolSelected()

		<code>slotDeleteProtocol()</code> <code>slotViewResult()</code> <code>slotExportResult()</code> <code>slotSelectedTreeItem()</code> <code>slotSubjetOrder()</code> <code>slotProtocolOrder()</code> <code>slotShowAll()</code> <code>slotShowResults()</code> <code>slotSelectResult()</code> <code>slotNextPage()</code> <code>slotPreviousPage()</code> <code>slotStartAnalysis()</code> <code>slotSelectAllFeatures()</code> <code>slotDeselectAllFeatures()</code> <code>slotDatasetSelected()</code> <code>slotExitFromAnalysis()</code> <code>slotContinueAnalysis()</code> <code>slotSelectOutputDirectory()</code>
--	--	---

**Tabella 12:** Descrizione test di unità

## E Pianificazione ed esecuzione del collaudo

Il software verrà collaudato sui seguenti sistemi operativi:

- Windows 7, senza Service Pack, o superiore sia a 32 che a 64 bit;
- Ubuntu 12.04 o superiore, sia a 32 che a 64 bit.

per verificare che soddisfi tutte le funzionalità richieste in modo corretto. Il collaudo verrà effettuato in due fasi distinte:

**Fase A:** il test verrà eseguito dai componenti del gruppo *Seven Monkeys* e consisterà nell'esecuzione del software al fine di soddisfare gli obiettivi fissati.

**Fase B:** come diretta conseguenza della fase precedente, in seguito all'esito positivo, il test verrà eseguito sul software in sede di collaudo con la presenza del committente.