



# Piano di Qualifica

## Informazioni sul documento

<b>Nome documento</b>	Piano di Qualifica
<b>Versione</b>	v2.0.0
<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>	Adami Alberto <i>Responsabile di Progetto</i>	2013-01-??	v2.0.0
<i>Eseguita verifica del documento</i>	Martignago Jimmy <i>Verificatore</i>	2014-02-03	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	2
2.2	Organizzazione	2
2.3	Pianificazione strategica e temporale	3
2.4	Responsabilità	3
2.5	Risorse necessarie	3
2.6	Tecniche, misure e metriche	4
2.6.1	Tecniche di analisi	4
2.6.2	Misure e metriche di qualità	4
<b>3</b>	<b>Gestione amministrativa della revisione</b>	<b>7</b>
3.1	Comunicazione e risoluzione di anomalie e discrepanze	7
3.2	Procedure di controllo di qualità di processo	7
3.3	Procedure di controllo di qualità di prodotto	8
<b>A</b>	<b>Standard di qualità</b>	<b>9</b>
A.1	Standard ISO/IEC 15504	9
A.2	Standard ISO/IEC 9126	10
<b>B</b>	<b>Resoconto delle attività di verifica</b>	<b>12</b>
B.1	Revisione dei Requisiti	12
B.2	Revisione di Progettazione	12
B.3	Revisione di Qualifica	12
B.4	Revisione di Accettazione	13
<b>C</b>	<b>Dettaglio dell'esito delle revisioni</b>	<b>14</b>
C.1	Revisione dei Requisiti	14
<b>D</b>	<b>Pianificazione dei test</b>	<b>15</b>
D.1	Test di sistema	15
D.1.1	Descrizione test di sistema	16
D.2	Test di integrazione	17
D.2.1	Descrizione test di integrazione	18
D.3	Test di validazione	20
D.3.1	Test TV1	20
D.3.2	Test TV2	20
D.3.3	Test TV3	20
D.3.4	Test TV4	20
D.3.5	Test TV5	20

D.3.6 Test TV6 . . . . .	21
D.3.7 Test TV7 . . . . .	21
D.3.8 Test TV8 . . . . .	21
D.3.9 Test TV9 . . . . .	21
D.3.10 Test TV10 . . . . .	21
D.3.11 Test TV11 . . . . .	22
D.3.12 Test TV12 . . . . .	22
D.4 Test di unità . . . . .	22
<b>E Pianificazione ed esecuzione del collaudo . . . . .</b>	<b>23</b>

## Elenco delle tabelle

3	Indice Gulpease dei documenti presentati in Revisione dei Requisiti .	12
4	Indice Gulpease dei documenti presentati in Revisione di Progettazione	12
5	Tracciamento test di sistema-requisiti . . . . .	16
6	Test di integrazione . . . . .	18
7	Tracciamento componente-test di integrazione . . . . .	19

## Elenco delle figure

1	Il ciclo PDCA . . . . .	8
2	Diagramma informale della strategia di integrazione . . . . .	17

# 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 obbiettivi qualitativi che si intendono applicare al prodotto da sviluppare. Per soddisfare tali obbiettivi, 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 v2.0.0* 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 v1.0.0*;
- **Capitolato d'appalto C3:** Romeo, Medical Image Cluster Analysis Tool <http://www.math.unipd.it/~tullio/IS-1/2013/Progetto/C3p.pdf>

### 1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v2.0.0*;
- **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>;
- **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>
- **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.

#### 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 v2.0.0*.
- **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 v2.0.0*), sarà rispettivamente il seguente:

- **Analisi (FA):** in tale fase si dovranno seguire i metodi di verifica descritti nelle *Norme di Progetto v2.0.0* sui documenti prodotti. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.1;

---

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

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

- **Analisi Incrementale (FAI):** 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 v2.0.0*. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.2;
- **Progettazione Architeturale (FPA):** 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 v2.0.0*. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.3;
- **Progettazione di Dettaglio e Codifica (FPDC):** 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 di Progettazione. Si dovranno rispettare inoltre le norme di codifica, descritte nelle *Norme di Progetto v2.0.0*. I risultati che emergeranno da tali attività, saranno descritti nell'appendice B.4;
- **Verifica Validazione (FVV):** 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 v2.0.0*. 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 v2.0.0*. 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 v2.0.0*, 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 v2.0.0*.

## 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 v2.0.0*.

Nelle *Norme di Progetto v2.0.0* 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*.

**Analisi statica:** è applicabile sia alla documentazione che al codice e permette di verificare la correttezza dei prodotti, individuandone errori e anomalie. Ci sono due tecniche per effettuare tale analisi: l'*analisi walkthrough* e l'*inspection*, entrambe descritte approfonditamente nelle *Norme di Progetto v2.0.0*.

**Analisi dinamica:** si applica solo al prodotto software e consiste nella verifica e validazione attraverso dei **test**. Per garantire la correttezza dei test è necessario che essi siano **ripetibili**: solamente dei test che, dato un certo input, producono sempre lo stesso output, riescono a verificare la correttezza del prodotto.

Prima di effettuare questo tipo di analisi, devono essere definiti:

- **Ambiente:** consiste sia del sistema software che del sistema hardware utilizzato;
- **Specifica dei valori:** consiste nel definire sia i parametri di input necessari per la corretta esecuzione del test, sia il valore atteso per i parametri di output;
- **Procedure:** dev'essere definito il modo in cui i test devono essere effettuati, specificando un eventuale ordine e come dovranno essere interpretati i risultati.

Di seguito saranno esposti i vari tipi di test che verranno utilizzati:

- **Test di unità:** verifica ogni singola unità del prodotto software, avvalendosi di strumenti come `driverG` e/o `stubG`. In particolare si verifica che per ogni unità siano esplorati tutti i cammini di esecuzione possibili e che i requisiti per quella determinata unità siano soddisfatti;
- **Test di integrazione:** per i moduli che hanno soddisfatto i test di unità, si procede alla verifica di più unità combinate, con lo scopo di evidenziare eventuali errori residui. Per effettuare tali test si farà uso di classi appositamente create per simulare e verificare l'interazione;
- **Test di sistema:** consiste nella validazione del sistema, attraverso la verifica delle coperture di tutti i requisiti individuati durante l'**Analisi Incrementale (FAI)**;
- **Test di regressione:** consiste nell'eseguire nuovamente i test di unità e di integrazione per le componenti che hanno subito modifiche, così da verificare che non siano stati introdotti nuovi errori e/o anomalie;
- **Test di accettazione:** consiste nel collaudo del sistema eseguito dal committente. Se l'esito di tale collaudo è positivo si può procedere al rilascio ufficiale del prodotto.

### 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 v2.0.0*), 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 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 per commento per il numero di linee di codice. Il range di accettazione è [ $>20$ ], quello ottimale è [ $>30$ ];
- **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  $\text{package}_G$  dell'architettura. Indica il numero di classi esterne al  $\text{package}_G$ , che hanno delle dipendenze con delle classi interne ad esso. Il valore di tale indice è direttamente proporzionale al grado di dipendenza del resto del software dal  $\text{package}_G$  preso in considerazione.

Tale indice dovrebbe avere un valore ragionevole. Se risulta troppo basso, può indicare che il  $\text{package}_G$  non offre sufficienti funzionalità al sistema e quindi può essere scarsamente utile. Se invece risulta troppo alto, potrebbe accadere che eventuali modifiche al  $\text{package}_G$ , 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  $\text{package}_G$  dell'architettura. Indica il numero di classi interne al  $\text{package}_G$ , che hanno delle dipendenze con delle classi esterne ad esso. Il valore di tale indice è direttamente proporzionale al grado di indipendenza del  $\text{package}_G$  preso in considerazione.

In generale, tale indice andrebbe tenuto basso. L'obiettivo è quello di aumentare le funzionalità del  $\text{package}_G$  senza duplicare i servizi che magari sono offerti da altri  $\text{package}_G$ .

## 3 Gestione amministrativa della revisione

### 3.1 Comunicazione e risoluzione di anomalie e discrepanze

Nel corso delle attività di verifica, si possono riscontrare delle anomalie e/o discrepanze.

- **Anomalia:** lo stato del prodotto non corrisponde a quelle che sono le sue aspettative; questo causa un totale malfunzionamento dell'intero sistema quindi, la gestione di tale problema assume una priorità alta nella gestione dei problemi;
- **Discrepanza:** si intende un allontanamento tra lo stato del prodotto attuale con le aspettative prefissate per quel prodotto; questo tipo di problema non comporta un malfunzionamento totale del sistema, ma lo rende inesatto al soddisfacimento dei requisiti specificati. Una discrepanza può essere vista come un'anomalia con priorità medio/bassa a seconda della gravità della stessa.

Per la gestione di questi problemi verrà utilizzato il sistema di ticketing offerto da GitHub<sup>3</sup>. Ogni qualvolta il verificatore riscontri un'anomalia, dovrà aprire un ticket apposito, riportante la descrizione dettagliata del problema.

Il *Responsabile di Progetto*, prima di approvare tale ticket, dovrà verificare che l'anomalia sussista e che questa non sia già compresa in altri ticket precedentemente approvati. Si riserva al Responsabile inoltre, la facoltà di non approvare tale ticket, se il costo per effettuare l'operazione di modifica, non fosse compensabile con il beneficio derivante dalla stessa. Le anomalie e le discrepanze potranno riguardare:

- Mancato rispetto con quanto indicato nelle *Norme di Progetto v2.0.0*;
- Mancato rispetto dei valori di accettazione da parte di una delle misurazioni (vedi 2.6.2);
- Allontanamento dai requisiti specificati durante l'analisi dei requisiti.

Si rimanda alla consultazione delle *Norme di Progetto v2.0.0* per le modalità di apertura di un ticket.

### 3.2 Procedure di controllo di qualità di processo

La qualità di un prodotto dipende dalla qualità del processo da cui proviene. Per garantire la qualità dei processi è necessario che i processi siano dettagliatamente pianificati e che ci sia un controllo continuo su di essi. Si è quindi deciso di aderire allo standard ISO<sub>9001</sub>/IEC<sub>15504</sub>, conosciuto anche con l'acronimo **SPICE** (Software Process Capability dEtermination)<sup>4</sup>.

La qualità dei processi verrà garantita inoltre, adottando il ciclo di Deming<sub>1946</sub> (PDCA). Il ciclo di Deming è un modello incrementale studiato per il miglioramento continuo della qualità dei processi. Si fonda sulla suddivisione in quattro fasi:

- **Plan:** si stabiliscono gli obiettivi e i processi necessari per raggiungere le aspettative prefissate;
- **Do:** si implementa il piano, eseguendo il processo, realizzando il prodotto;
- **Check:** si studiano i risultati ottenuti comparandoli con i risultati attesi;
- **Act:** si procede alla correzione dei processi, determinando la causa della discrepanza tra i risultati ottenuti e quelli attesi.

<sup>3</sup>Per una descrizione dettagliata fare riferimento a documento *Norme di Progetto v2.0.0*

<sup>4</sup>Per una descrizione dettagliata dello standard, consultare l'appendice A.1



**Figura 1:** Il ciclo PDCA

### 3.3 Procedure di controllo di qualità di prodotto

Per controllare la qualità del prodotto saranno svolte le seguenti attività:

- **Quality assurance:** si verifica che il prodotto soddisfi gli obiettivi di qualità prefissati;
- **Verifica:** attività svolta durante l'intera durata del progetto, ha come scopo quello di controllare che il prodotto corretto e congruo alle aspettative;
- **Validazione:** viene verificato che il prodotto risponda ai requisiti specificati.

## 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

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 3:** 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

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 4:** 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

Questa sezione verrà redatta non appena gli esiti delle attività di verifica per la rispettiva revisione, saranno disponibili.

#### **B.4 Revisione di Accettazione**

Questa sezione verrà redatta non appena gli esiti delle attività di verifica per la rispettiva revisione, saranno disponibili.

## 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.

## 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 v2.0.0* 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 v2.0.0*.

### 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 v2.0.0*. Vengono qui di seguito riportati i test di sistema ritenuti meritevoli di un test, al fine di garantire il soddisfacimento dei requisiti software individuati.

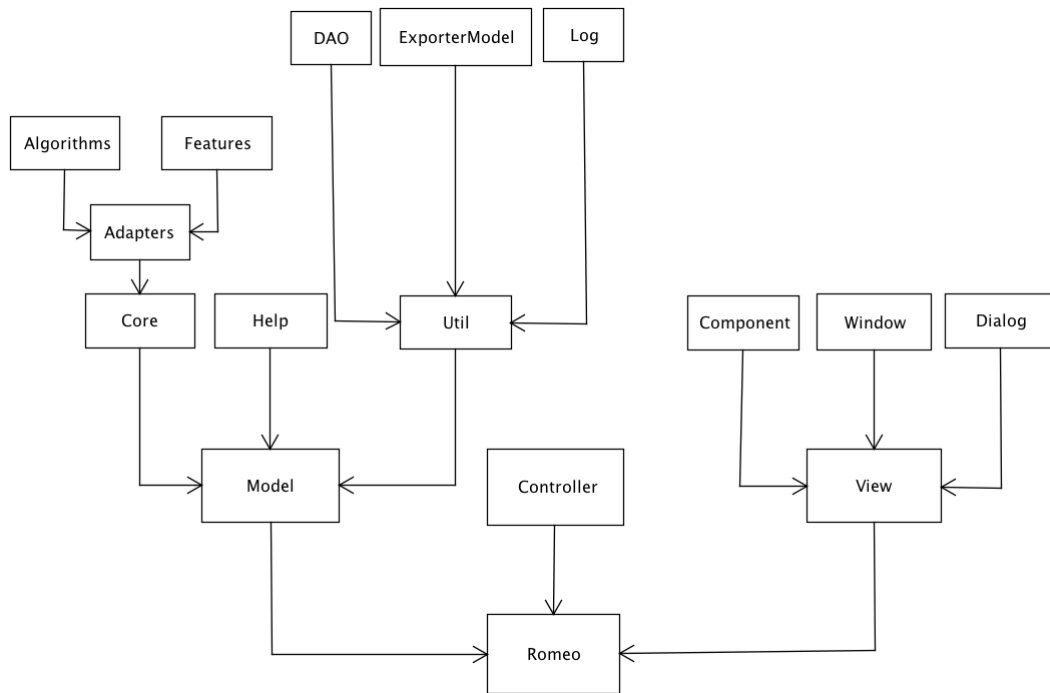
### D.1.1 Descrizione test di sistema

Test	Descrizione	Stato	Requisito
TS1.2.1.1	Viene verificato che il sistema accetti in input file di formato PNG	N.I.	R0F1.2.1.1
TS1.2.1.2	Viene verificato che il sistema accetti in input file di formato JPG	N.I.	R0F1.2.1.2
TS1.2.1.3	Viene verificato che il sistema accetti in input file di formato BMP	N.I.	R0F1.2.1.3
TS1.2.1.4	Viene verificato che il sistema accetti in input file di formato AVI	N.I.	R0F1.2.1.4
TS1.2.1.5	Viene verificato che il sistema accetti in input file di formato NIfTI	N.I.	R0F1.2.1.5
TS1.2.1.6	Viene verificato che il sistema accetti in input file di formato Analyze7.5	N.I.	R0F1.2.1.6
TS1.3.1	Viene verificato che il sistema accetti in input file di formato PNG per le maschere	N.I.	R0F1.3.1
TS1.3.2	Viene verificato che il sistema accetti in input file di formato JPG per le maschere	N.I.	R0F1.3.2
TS1.3.3	Viene verificato che il sistema accetti in input file di formato BMP per le maschere	N.I.	R0F1.3.3
TS1.3.4	Viene verificato che il sistema accetti in input file di formato NIfTI per le maschere	N.I.	R0F1.3.4
TS1.3.5	Viene verificato che il sistema accetti in input file di formato Analyze7.5 per le maschere	N.I.	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	N.I.	R0F10.2.1
TS12.2	Viene verificato che i risultati delle analisi effettuate vengano esportati nel formato corretto	N.I.	R0F12.2
TS13.1	Viene verificato che vengano visualizzate correttamente le immagini 2D all'interno del software	N.I.	R0F13.1
TS13.2	Viene verificato che vengano visualizzate correttamente le immagini 3D all'interno del software	N.I.	R0F13.2
TS16	Viene verificato che il sistema funzioni su Windows 7 32bit senza service pack o superiore	N.I.	R0V16
TS17	Viene verificato che il sistema funzioni su Windows 7 64bit senza service pack o superiore	N.I.	R0V17
TS18	Viene verificato che il sistema funzioni su Ubuntu 12.04 32bit o superiore	N.I.	R0V18
TS19	Viene verificato che il sistema funzioni su Ubuntu 12.04 64bit o superiore	N.I.	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	N.I.	R0V21

**Tabella 5:** 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 2:** Diagramma informale della strategia di integrazione



### D.2.1 Descrizione test di integrazione

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

**Tabella 6:** Test di integrazione

Componente	Test
Romeo	TI-Romeo
Romeo::Controller	Architettura del sistema
Romeo::Model	TI-Model
Romeo::Model::Core	TI-Core
Romeo::Model::Core::Adapters	TI-Adapters
Romeo::Model::Core::Adapters::Algorithms	TI-Algorithms
Romeo::Model::Core::Adapters::Features	TI-Features
Romeo::Model::Help	TI-Help
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::View	TI-View
Romeo::View::Component	TI-Component
Romeo::View::Dialog	TI-Dialog
Romeo::View::Window	TI-Window

**Tabella 7:** 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 v2.0.0*.

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

Questa sezione verrà redatta durante la fase di progettazione di dettaglio.

## E Pianificazione ed esecuzione del collaudo

Questa sezione verrà redatta non appena il prodotto sviluppato dal gruppo *Seven Monkeys* sarà prossimo al collaudo del sistema.