



Piano di Qualifica

Informazioni sul documento

Nome documento	Piano di Qualifica
Versione	v1.0.0
Data redazione	2013-12-09
Redattori	<ul style="list-style-type: none">• Adami Alberto• Bissacco Nicolò• Magnabosco Nicola
Verificatori	<ul style="list-style-type: none">• Scapin Davide
Approvazione	<ul style="list-style-type: none">• Feltre Beatrice
Lista distribuzione	<ul style="list-style-type: none">• 7Monkeys• Prof. Tullio Vardanega• Prof. Riccardo Cardin• Departement of Information Engineering (DEI)
Uso	Esterno

Sommario

Documento riguardante le strategie di verifica e validazione adottate dal gruppo 7Monkeys volte al soddisfacimento dei requisiti qualitativi nello svolgimento del progetto Romeo

Diario delle Modifiche

Modifica	Autore & Ruolo	Data	Versione
<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

1	Introduzione	1
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
2	Visione generale della strategia di verifica	2
2.1	Organizzazione	2
2.2	Pianificazione strategica e temporale	2
2.3	Responsabilità	2
2.4	Risorse necessarie	2
2.5	Tecniche, misure e metriche	3
2.5.1	Tecniche di analisi	3
2.5.2	Misure e metriche di qualità	4
3	Gestione amministrativa della revisione	5
3.1	Comunicazione e risoluzione di anomalie e discrepanze	5
3.2	Procedure di controllo di qualità di processo	5
3.3	Procedure di controllo di qualità di prodotto	6
A	Standard di qualità	7
A.1	Standard ISO/IEC 15504	7
A.2	Standard ISO/IEC 9126	8
B	Resoconto delle attività di verifica	10
B.1	Revisione dei Requisiti	10
B.2	Revisione di Progettazione	10
B.3	Revisione di Qualifica	10
B.4	Revisione di Accettazione	10
C	Pianificazione ed esecuzione del collaudo	11



Elenco delle tabelle

3	Indice Gulpease, Analisi	10
---	------------------------------------	----



Elenco delle figure

1	Il ciclo PDCA	5
---	-------------------------	---

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 che permetterà di applicare la *Cluster Analysis*_G a dati biomedici. 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*_G.

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 v1.0.0* 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 v1.0.0*;
- **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/swebok/html/ch11>;
- **Standard ISO_G/IEC_G 9126:** Software Engineering - Product Quality http://en.wikipedia.org/wiki/ISO/IEC_9126
- **Standard ISO_G/IEC_G 15504 (SPICE):** Information Technology — Process assessment
 - http://en.wikipedia.org/wiki/ISO/IEC_15504;
 - <http://www.ehealthinformation.ca/documents/1072.pdf>
- **Indice Gulpease:** http://it.wikipedia.org/wiki/Indice_Gulpease.

2 Visione generale della strategia di verifica

2.1 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 v1.0.0*), sarà rispettivamente il seguente:

- **Analisi (FA):** in tale fase si dovranno seguire i metodi di verifica descritti nelle *Norme di Progetto v1.0.0* sui documenti prodotti. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.1;
- **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 v1.0.0*. I risultati che emergeranno da tali attività di verifica, saranno descritti nell'appendice B.2;
- **Progettazione Architettuale (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 v1.0.0*. 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 v1.0.0*. 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 v1.0.0*. Gli esiti di tali attività verranno descritti nell'appendice C.

2.2 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 v1.0.0*. 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 v1.0.0*, dove saranno descritte le varie metodologie da seguire per l'individuazione e la correzione degli errori.

2.3 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 v1.0.0*.

2.4 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 v1.0.0*.

Nelle *Norme di Progetto v1.0.0* 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.

2.5 Tecniche, misure e metriche

2.5.1 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*.

- **Walkthrough:** questa tecnica di analisi prevede una lettura critica del codice o del documento prodotto. Tale tecnica è molto dispendiosa in termini di risorse, visto che viene applicata all'intero documento, senza avere una precisa idea di quale sia il tipo di anomalia e di dove ricercarla. Essa è però necessaria nelle prime fasi del progetto, vista l'inesperienza da parte del gruppo nell'attuare un tipo di verifica più precisa e mirata. Dopo una prima fase di lettura ed identificazione degli errori, si procede alla discussione degli stessi, proponendo le modifiche da apportare per garantirne la correzione. Il passo finale consiste nell'applicare le modifiche proposte, redigendo un rapporto preciso che elenchi le modifiche effettuate. Una caratteristica di questo tipo di analisi è che richiede l'utilizzo di più risorse umane;
- **Inspection:** questa tecnica di analisi presuppone l'esperienza da parte del verificatore, nell'individuare gli errori e le anomalie più frequenti. A tal scopo è necessaria una *lista di controllo*, stilata in una precedente analisi di tipo walkthrough, nella quale vengono elencate le sezioni più critiche. Questo permette quindi una verifica più rapida che non necessiti della lettura dell'intero documento o codice, oltre che richiedere meno risorse umane. Dopo aver terminato l'analisi, sarà necessario stilare un rapporto di verifica che tenga traccia del lavoro svolto e delle modifiche apportate.

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 driver_G e/o stub_G. 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.

2.5.2 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 v1.0.0*), 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].

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 GitHub¹. 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 v1.0.0*;
- Mancato rispetto dei valori di accettazione da parte di una delle misurazioni (vedi 2.5.2);
- Allontanamento dai requisiti specificati durante l'analisi dei requisiti.

Si rimanda alla consultazione delle *Norme di Progetto v1.0.0* 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_G/IEC_G15504, conosciuto anche con l'acronimo **SPICE** (Software Process Capability dEtermination)².

La qualità dei processi verrà garantita inoltre, adottando il ciclo di Deming_G (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.



Figura 1: Il ciclo PDCA

¹Per una descrizione dettagliata fare riferimento a documento *Norme di Progetto v1.0.0*

²Per una descrizione dettagliata dello standard, consultare l'appendice A.1

3.3 Procedure di controllo di qualità di prodotto

Il gruppo intende aderire allo standard ISO_G/ IEC_G9126, al fine di garantire la qualità del prodotto da realizzare. 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*.

Si faccia riferimento all'appendice A.2 per una descrizione dettagliata dello standard.

A Standard di qualità

A.1 Standard ISO/IEC 15504

Lo standard ISO_G/IEC_G15504, 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*_G 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_G/IEC_G9126 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.

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, Analisi

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: Tutti i documenti verificati, soddisfano i range di accettazione specificati nella sezione 2.5.2. 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

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

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 Pianificazione ed esecuzione del collaudo

Questa sezione verrà redatta non appena il prodotto sviluppato dal gruppo **7Monkeys** sarà prossimo al collaudo del sistema.