

# SiVoDiM

---

## Sintesi Vocale per Dispositivi Mobili



### Piano di Qualifica

Versione	1.0.0
Redattori	Federico Rossetto
Verificatori	Riccardo Rizzo
Responsabili	Enrico Chiara
Uso	Esterno
Lista di distribuzione	Stark Labs Prof. Tullio Vardanega, Prof. Riccardo Cardin

Documento riguardante l'insieme di strategie di verifica adottate dal gruppo Stark Labs per il conseguimento di requisiti qualitativi per il progetto SiVoDiM

## Registro delle modifiche

Attività	Autori	Data	Versione
Accettazione	Enrico Chiara	31/03/2016	1.0.0
Verifica	Riccardo Rizzo	31/03/2016	0.2.0
Calcolo indice Gulpease dei documenti	Federico Rossetto	31/03/2016	0.2.0
Correzione errori	Federico Rossetto	24/03/2016	0.1.1
Verifica	Riccardo Rizzo	23/03/2016	0.1.0
Stesura standard di qualità	Federico Rossetto	19/03/2016	0.0.4
Stesura gestione amministrativa	Federico Rossetto	18/03/2016	0.0.3
Stesura strategie di verifica	Federico Rossetto	12/03/2016	0.0.2
Stesura della struttura	Federico Rossetto	11/03/2016	0.0.1

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento . . . . .	1
1.2	Scopo del progetto . . . . .	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>3</b>
2.1	Definizione obiettivi . . . . .	3
2.1.1	Qualità di processo . . . . .	3
2.1.2	Qualità di prodotto . . . . .	3
2.2	Procedure di controllo di qualità di processo . . . . .	3
2.3	Procedure di controllo della qualità del prodotto . . . . .	3
2.4	Organizzazione . . . . .	4
2.5	Pianificazione strategica e temporale . . . . .	4
2.6	Responsabilità . . . . .	4
2.7	Risorse . . . . .	4
2.8	Tecniche di analisi . . . . .	4
2.8.1	Analisi statica . . . . .	4
2.8.1.1	Walkthrough . . . . .	5
2.8.1.2	Inspection . . . . .	5
2.8.2	Analisi dinamica . . . . .	5
2.8.2.1	Test di unità . . . . .	5
2.8.2.2	Test di integrazione . . . . .	5
2.8.2.3	Test di sistema . . . . .	6
2.8.2.4	Test di regressione . . . . .	6
2.8.2.5	Test di accettazione . . . . .	6
2.9	Misure e metriche . . . . .	6
2.9.1	Metriche per i processi . . . . .	6
2.9.1.1	Schedule Variance (SV) . . . . .	6
2.9.1.2	Budget Variance (BV) . . . . .	6
2.9.2	Metriche per i documenti . . . . .	7
2.9.2.1	Gulpease <sub>G</sub> . . . . .	7
2.9.3	Metriche per il software . . . . .	7
2.9.3.1	Numero di livelli di annidamento . . . . .	7
2.9.3.2	Numero di attributi per classe . . . . .	7
2.9.3.3	Numero di parametri per metodo . . . . .	7
2.9.3.4	Linee di codice per linee di commento . . . . .	8
2.9.3.5	Copertura di codice . . . . .	8
<b>3</b>	<b>Gestione amministrativa della revisione</b>	<b>9</b>
3.1	Comunicazione e risoluzione di anomalie . . . . .	9

<b>4</b>	<b>Standard di qualità</b>	<b>10</b>
4.1	Standard ISO/IEC 15504 . . . . .	10
4.2	Ciclo di Deming . . . . .	11
4.3	Standard ISO/IEC 9126 . . . . .	12
<b>5</b>	<b>Resoconto delle attività di verifica</b>	<b>14</b>
5.1	Riassunto delle attività di verifica . . . . .	14
5.1.1	Revisione dei Requisiti . . . . .	14
5.2	Dettaglio delle verifiche tramite analisi . . . . .	14
5.2.1	Documenti . . . . .	14
5.2.2	Processi . . . . .	15
5.2.3	Esito delle revisioni . . . . .	15

## Elenco delle tabelle

1	Indici Gulpease dei documenti . . . . .	14
2	Metriche dei processi . . . . .	15

## Elenco delle figure

1	Ciclo di Deming . . . . .	11
---	---------------------------	----

## 1 Introduzione

### 1.1 Scopo del documento

Questo documento contiene le strategie che il gruppo di lavoro ha deciso di adottare per raggiungere gli obiettivi qualitativi nel progetto **SiVoDiM**. All'interno di tale ottica, è necessario un continuo processo di verifica, con lo scopo di correggere tempestivamente e senza spreco di risorse le anomalie riscontrate sulle attività svolte.

### 1.2 Scopo del progetto

Lo scopo del progetto risiede nello sviluppo di un'applicazione utile a dimostrare efficacemente le potenzialità del motore di sintesi vocale FA-TTS<sub>G</sub> realizzato dall'azienda MIVOQ s.r.l. e messo a disposizione del gruppo di lavoro. Si devono realizzare due applicazioni per sistemi Android<sub>G</sub>:

- **Applicazione di configurazione:** deve permettere all'utente di interfacciarsi direttamente con il sistema operativo per configurare, salvare e modificare le voci ereditate dal motore di sintesi FA-TTS di MIVOQ;
- **Applicazione per la creazione di sceneggiati:** permette la creazione e il salvataggio di racconti e sceneggiati, che possono essere esportati in formato audio attraverso l'utilizzo del motore FA-TTS.

Entrambe le applicazioni devono interfacciarsi con due moduli di basso livello:

- **Modulo di sistema:** permette di interfacciarsi tramite connessione di rete al motore FA-TTS;
- **Libreria:** una libreria contenente tutte le funzionalità offerte dal motore FA-TTS, utile nell'ottica di un riuso futuro del *software*.

Lo sviluppo di tutte e quattro le suddette componenti è a carico del gruppo Stark Labs.

### 1.3 Glossario

Al fine di aumentare la comprensione del testo ed evitare eventuali ambiguità, viene fornito un glossario (*Glossario v1.0.0*) contenente le definizioni degli acronimi e dei termini tecnici utilizzati nel documento. Ogni vocabolo contenuto nel glossario è contrassegnato dal pedice "<sub>G</sub>".

### 1.4 Riferimenti

#### 1.4.1 Normativi

- *Norme di Progetto v1.0.0*;
- **Capitolato C6 – SiVoDiM:** Sintesi Vocale per Dispositivi Mobili.

#### 1.4.2 Informativi

- *Glossario v1.0.0*;
- *Piano di progetto v1.0.0*;

- Slide dell'insegnamento Ingegneria del Software modulo A:
  - Ingegneria dei requisiti;
  - Diagrammi dei casi d'uso  
<http://www.math.unipd.it/~tullio/IS-1/2015/>.
- Software Engineering - Ian Sommerville - 9th Edition 2010:
  - Chapter 24: Quality management;
  - Chapter 25: Configuration management;
  - Chapter 26: Process improvement.
- SWEBOK - Version 3 (2004): capitolo 11 - Software Quality;
- IEEE 830-1998: [https://en.wikipedia.org/wiki/Software\\_requirements\\_specification](https://en.wikipedia.org/wiki/Software_requirements_specification);
- 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

Di seguito vengono descritti gli obiettivi di qualità relativi al prodotto commissionato e ai processi necessari per un sua corretta realizzazione.

#### 2.1.1 Qualità di processo

Per garantire la qualità del prodotto si deve perseguire la qualità dei processi che lo definiscono. Per tale ragione si è deciso di utilizzare lo standard ISO<sub>G</sub>/IEC<sub>G</sub> 15504 noto come SPICE, che fornisce gli strumenti necessari per valutare l'idoneità dei processi. Per applicare correttamente il suddetto modello si deve far uso del ciclo di *Deming<sub>G</sub>*, che definisce una metodologia di controllo specifica per i processi nel corso del loro ciclo di vita, con l'intento di migliorarne costantemente la qualità.

#### 2.1.2 Qualità di prodotto

Per aumentare il valore commerciale di un prodotto *software*, e per garantire il corretto funzionamento dello stesso, è necessario fissare degli obiettivi qualitativi e verificare che questi vengano rispettati. Lo standard ISO<sub>G</sub>/IEC<sub>G</sub> 9126 è stato redatto allo scopo di definire questi obiettivi e delineare alcune metriche capaci di misurare il raggiungimento degli stessi.

### 2.2 Procedure di controllo di qualità di processo

La qualità dei processi è garantita dalla corretta applicazione del metodo PDCA<sub>G</sub>. Grazie a questo strumento si può garantire un continuo miglioramento della qualità dei processi, inclusa la verifica, con un conseguente miglioramento dei prodotti creati. Per avere il controllo dei processi e della qualità è necessario che:

- I processi siano pianificati dettagliatamente;
- Vengano ripartite chiaramente le risorse nella pianificazione;
- Ci sia controllo sui processi.

L'attuazione di tali punti è descritta dettagliatamente nel *Piano di Progetto v1.0.0*. La qualità dei processi viene controllata tramite l'analisi costante della qualità del prodotto. Un prodotto di bassa qualità è indice di un processo che deve essere migliorato. Per quantificare la qualità dei processi vengono utilizzate le metriche descritte nella sezione 2.9 Misure e metriche.

### 2.3 Procedure di controllo della qualità del prodotto

Si garantisce il controllo della qualità del prodotto attraverso il rispetto dei seguenti punti:

- **Quality assurance:** è l'insieme di attività svolte al fine di garantire il raggiungimento degli obiettivi di qualità. Prevede l'attuazione di tecniche di analisi statica e dinamica, descritte nella sezione 2.8.1 Analisi statica;
- **Verifica:** processo che determina se il risultato di una fase è corretto. La verifica viene eseguita costantemente durante l'intera durata del progetto. I risultati delle attività di verifica eseguiti nelle varie fasi di progetto sono riportate nella sezione 5 Resoconto delle attività di verifica;

- **Validazione:** la conferma oggettiva che il sistema risponda ai requisiti.

## 2.4 Organizzazione

L'organizzazione della strategia di verifica è basata sull'utilizzo di attività di controllo per ogni processo attuato. Per ognuno di questi viene verificata la qualità ed eventualmente si verifica anche la qualità del prodotto ottenuto. Ognuna delle fasi del progetto descritte nel *Piano di Progetto v1.0.0* necessita di diverse attività di verifica dipendenti da differenti output:

- **Analisi:** in questa fase è necessario seguire i metodi di verifica descritti nella sezione 2.8 Tecniche di analisi sui documenti prodotti. La realizzazione di tali attività di verifica sono descritte nella sezione 4.1 Standard ISO/IEC 15504.

In ogni documento viene inoltre incluso il registro delle modifiche che permette di mantenere uno storico delle attività svolte e delle relative responsabilità.

## 2.5 Pianificazione strategica e temporale

Dato che l'obiettivo è di rispettare le scadenze fissate nel *Piano di Progetto v1.0.0*, è necessario che l'attività di verifica sia ben organizzata. Pertanto l'individuazione e la correzione di errori dovrà essere tempestiva, in modo da impedire che si diffondano. Ogni attività di redazione di documenti o di codifica deve essere preceduta da un'analisi della struttura e dei contenuti. Questo allo scopo di evitare imprecisioni concettuali o tecniche e rendendo l'attività di verifica più semplice, con conseguente riduzione del numero di correzioni richieste. La metodologia da seguire per individuare e correggere eventuali errori è descritta nelle *Norme di Progetto v1.0.0*.

## 2.6 Responsabilità

Per garantire che il processo di verifica sia efficace e sistematico, vengono attribuite tali responsabilità a *Verificatore* e *Responsabile di Progetto*. La suddivisione dei compiti e le modalità sono definite nelle *Norme di Progetto v1.0.0*.

## 2.7 Risorse

Per assicurarsi che gli obiettivi vengano raggiunti sono necessarie risorse umane e tecnologiche. Coloro che detengono la maggiore responsabilità per le attività di verifica e validazione sono *Verificatore* e *Responsabile di Progetto*. I ruoli sono descritti nel dettaglio nelle *Norme di Progetto v1.0.0*. Per risorse tecniche e tecnologiche sono intesi tutti gli strumenti *software* e *hardware* che il gruppo intende utilizzare. Affinché il lavoro dei *Verificatori* venga semplificato, sono stati impostati alcuni strumenti di controllo sistematico. Questi sono descritti in modo accurato nelle *Norme di Progetto v1.0.0*.

## 2.8 Tecniche di analisi

### 2.8.1 Analisi statica

Per analisi statica si intende una tecnica di controllo che permette di effettuare la verifica di quanto prodotto individuando eventuali errori. Essa viene svolta in due modi complementari.

**2.8.1.1 Walkthrough** Viene svolta una lettura critica di tutto il materiale. Questa tecnica è utile nelle prime fasi di progetto, quando i membri del gruppo non hanno ancora un'adeguata esperienza che permette verifiche più mirate. Grazie a questa tecnica, il *Verificatore* può stilare una lista degli errori più frequenti, in modo da migliorare le attività di analisi future. Questa attività è onerosa e richiede l'intervento di più persone per essere efficace ed efficiente. In seguito alla lettura segue una fase di discussione con il fine di esaminare i difetti e proporre le correzioni. La fase finale consiste nello stilare un rapporto che elenchi le modifiche effettuate.

**2.8.1.2 Inspection** In questa tecnica viene eseguita un'analisi mirata delle parti del documento o del codice che sono ritenute maggiormente fonte di errore. La lista di controllo, che contiene queste sezioni critiche, è redatta anticipatamente ed è frutto dell'esperienza dei *Verificatori* in seguito all'applicazione del Walkthrough. Questa strategia è più rapida del Walkthrough in quanto riduce il numero di parti da analizzare. Per tale ragione l'Inspection può essere eseguita solamente dai *Verificatori*, che individuano e correggono eventuali errori e redigono il rapporto di verifica per tracciare il lavoro svolto.

## 2.8.2 Analisi dinamica

L'analisi dinamica viene applicata solamente alla produzione di codice e viene effettuata durante l'esecuzione mediante l'uso di test utilizzati per verificarne il funzionamento. Per rendere questa attività utile e generare risultati attendibili, è necessario che i test siano ripetibili. Questo significa che il programma da un determinato *input* generi sempre lo stesso *output*. Test di questo tipo sono utili per determinare la correttezza ed evidenziare eventuali problemi in un *software*. Devono quindi essere definiti:

- **Ambiente:** consiste sia del sistema *hardware* che di quello *software* sui quali è pianificato lo sviluppo. Di questi è necessario specificare lo stato iniziale dal quale iniziare i test;
- **Specifiche:** consiste nel definire gli *input* e i rispettivi *output* attesi;
- **Procedure:** consiste nella definizione di come i test vengono svolti, con quale ordine e come vengono analizzati i risultati.

Esistono cinque tipi diversi di test: test di unità, test di integrazione, test di sistema, test di regressione e test di accettazione.

**2.8.2.1 Test di unità** Consta nel verificare ogni singola unità del *software* con l'utilizzo di *stub<sub>G</sub>*, *driver<sub>G</sub>* e *logger<sub>G</sub>*. Con unità è inteso il minimo quantitativo di *software*, che sia utile verificare singolarmente, prodotto da un singolo programmatore. Con questi test si verifica il funzionamento corretto dei moduli per eliminare dal sistema possibili errori di implementazione.

**2.8.2.2 Test di integrazione** Consta nel verificare i componenti del sistema che vengono aggiunti, al fine di analizzare che la combinazione di due o più unità *software* funzionino come previsto. L'obiettivo di questo test è di individuare errori residui dalla realizzazione dei moduli o comportamenti inaspettati da componenti *software* forniti da terze parti. Per effettuare questi test è necessario aggiungere delle componenti fittizie per sostituire quelle ancora non sviluppate, al fine di non influenzare l'esito dell'analisi.

**2.8.2.3 Test di sistema** Consta nel validare il prodotto *software* nel momento in cui si ritiene che sia giunto ad una versione definitiva. Questo test verifica che tutti i requisiti software stabiliti nell'*Analisi dei Requisiti v1.0.0* vengano rispettati.

**2.8.2.4 Test di regressione** Consta nell'eseguire nuovamente i test sulle componenti *software* in seguito a nuove modifiche, al fine di controllare che i cambiamenti non alterino il corretto funzionamento di queste componenti o di altre che non sono state aggiornate. Questa operazione viene facilitata dal tracciamento, che permette di individuare e ripetere i test di unità, integrazione e di sistema che sono stati influenzati dalla modifica.

**2.8.2.5 Test di accettazione** Consta nel collaudare il prodotto *software* in presenza del Proponente. Se questo collaudo viene superato, si può procedere al rilascio ufficiale del prodotto sviluppato.

## 2.9 Misure e metriche

Il processo di verifica deve essere quantificabile per risultare informativo. Pertanto, è necessario stabilire a priori delle metriche su cui basare le misurazioni del processo di verifica. Essendo le metriche di natura variabile, vengono definite due tipologie di intervalli:

- **Accettazione:** valori che vengono richiesti affinché il prodotto sia accettato;
- **Ottimale:** valori entro cui è consigliabile che la misurazione si collochi.

Non sono intervalli vincolanti, ma consigliati. Se tali valori si discostano, è necessaria una verifica approfondita.

### 2.9.1 Metriche per i processi

Come metriche per valutare i processi si è scelto di utilizzare degli indici che analizzino i costi e i tempi.

**2.9.1.1 Schedule Variance (SV)** Valuta se si è in linea, in anticipo o in ritardo rispetto alla pianificazione della *baseline<sub>G</sub>*. Questo è un indice di efficacia. Se  $SV > 0$  significa che il team sta producendo con maggior velocità rispetto alla pianificazione, viceversa se è negativo.

**Parametri utilizzati:**

- Range di accettazione:  $[> -(\text{Costo preventivo fase} \times 5\%)]$ ;
- Range ottimale:  $[>0]$ .

**2.9.1.2 Budget Variance (BV)** Valuta nella data corrente la differenza tra la spesa attuale e il costo pianificato. Se  $BV > 0$  significa che il progetto sta consumando il budget più lentamente di quanto pianificato, viceversa se negativo.

**Parametri utilizzati:**

- Range di accettazione:  $[> -(\text{Costo preventivo fase} \times 10\%)]$ ;
- Range ottimale:  $[>0]$ .

### 2.9.2 Metriche per i documenti

Come metrica per i documenti si è scelto di utilizzare un indice di leggibilità. L'indice utilizzato è specifico per la lingua italiana.

**2.9.2.1 Gulpease<sub>G</sub>** L'indice Gulpease<sub>G</sub> è un indice di leggibilità tarato sulla lingua italiana. Viene preferito rispetto ad altri poiché utilizza la lunghezza in lettere anziché in sillabe, semplificandone il calcolo. Questo indice evidenzia la complessità dello stile del documento. L'indice è calcolato secondo la seguente formula:

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

I risultati sono compresi tra 0 e 100, con 100 la migliore leggibilità e 0 la peggiore.

**Parametri utilizzati:**

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

### 2.9.3 Metriche per il software

È desiderabile, per poter raggiungere gli obiettivi di qualità del *software*, applicare le metriche a seguire.

**2.9.3.1 Numero di livelli di annidamento** Tale indice rappresenta il numero di livelli di annidamento dei metodi, ossia l'annidamento delle strutture. Un alto livello di questo indice può essere sintomo di un'elevata complessità del codice o di un basso livello di astrazione.

**Parametri utilizzati:**

- Range di accettazione: [1-6];
- Range ottimale: [1-3].

**2.9.3.2 Numero di attributi per classe** Tale indice rappresenta il numero di attributi contenuti in una classe. Un valore elevato può indicare la necessità di suddividere la classe in più classi. Pertanto, tale valore potrebbe essere sintomo di errori progettuali.

**Parametri utilizzati:**

- Range di accettazione: [0-16];
- Range ottimale: [3-8].

**2.9.3.3 Numero di parametri per metodo** Tale indice rappresenta il numero di parametri contenuti nei metodi. Un valore elevato potrebbe indicare un metodo con funzionalità eccessivamente complesse, sintomo di errori progettuali.

**Parametri utilizzati:**

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

**2.9.3.4 Linee di codice per linee di commento** Tale indice viene calcolato come il rapporto tra le linee di commento e le linee di codice. Valori ottimali di questo parametro indicano un codice più manutenibile.

**Parametri utilizzati:**

- Range di accettazione: [ $>0.25$ ];
- Range ottimale: [ $>0.30$ ].

**2.9.3.5 Copertura di codice** Tale indice indica la percentuale di istruzioni eseguite durante i test. Maggiore è questo parametro, minore sarà la probabilità di errori nel codice. Questo valore può essere abbassato dalla presenza di metodi semplici, come *setter* o *getter*.

**Parametri utilizzati:**

- Range di accettazione: [42%-100%];
- Range ottimale: [65%-100%].

## 3 Gestione amministrativa della revisione

### 3.1 Comunicazione e risoluzione di anomalie

Un'anomalia viene identificata da almeno uno dei seguenti casi:

- Violazione delle norme tipografiche di un documento;
- Violazione dei range di accettazione da parte degli indici di misurazione, descritti nella sezione 2.9 Misure e metriche;
- Incongruenze tra il codice e il *design* del prodotto;
- Incongruenze delle funzionalità del prodotto con funzionalità indicate nel documento *Analisi dei Requisiti v1.0.0*.

Se un *Verificatore* individua un'anomalia, dovrà aprire un nuovo *task* nell'apposita sezione Task di *Teamwork<sub>G</sub>*, come descritto nelle *Norme di Progetto v1.0.0*.

## 4 Standard di qualità

### 4.1 Standard ISO/IEC 15504

Questo standard descrive come i processi debbano essere controllati costantemente, al fine di rilevare possibili rischi intrinseci che potrebbero impedire di raggiungere gli obiettivi prefissati. I risultati delle singole valutazioni devono essere ripetibili, oggettivi e comparabili per far sì che contribuiscano al miglioramento effettivo dei processi in esame.

Il modello SPICE<sub>G</sub> definisce sei livelli di maturità del processo:

- **Incomplete process:** il processo non viene attuato o non raggiunge i risultati previsti;
- **Performed:** il processo viene completato e raggiunge i risultati previsti
  - **Process performance attribute:** capacità di raggiungere gli obiettivi.
- **Managed process:** il processo viene eseguito in maniera controllata a seconda degli obiettivi
  - **Performance management attribute:** è la capacità di un processo di elaborare un prodotto coerente con gli obiettivi;
  - **Work product management attribute:** è la capacità di un processo di elaborare un prodotto documentato, controllato e verificato.
- **Established process:** il processo viene eseguito in base a principi dell'Ingegneria del Software
  - **Process definition attribute:** l'esecuzione di processo si basa sugli standard per raggiungere gli obiettivi;
  - **Process resource attribute:** capacità del processo di utilizzare le risorse tecniche e umane appropriate, per essere efficaci.
- **Predictable process:** il processo viene eseguito costantemente in limiti predefiniti al fine di raggiungere i risultati attesi
  - **Measurement attribute:** gli obiettivi e le misure dei prodotti e dei processi sono utilizzati per garantire il raggiungimento degli obiettivi;
  - **Process control attribute:** il processo è controllato grazie alle misure di prodotto e processo, per effettuare correzioni migliorative.
- **Optimizing process:** il processo cambia e si adatta costantemente per raggiungere gli obiettivi
  - **Process change attribute:** i cambiamenti strutturali, di gestione e di esecuzione sono gestiti in maniera controllata al fine di raggiungere gli obiettivi;
  - **Continuous improvement attribute:** ogni modifica ai processi è identificata e implementata per garantire il miglioramento nella realizzazione degli obiettivi di business.

Ogni attributo di processo descritto è misurabile e lo standard predispone quattro livelli:

- **Non posseduto:** 0% - 15%;



- **Parzialmente posseduto:** 15% - 50%;
- **Largamente posseduto:** 50% - 85%;
- **Completamente posseduto:** 85% - 100%.

## 4.2 Ciclo di Deming

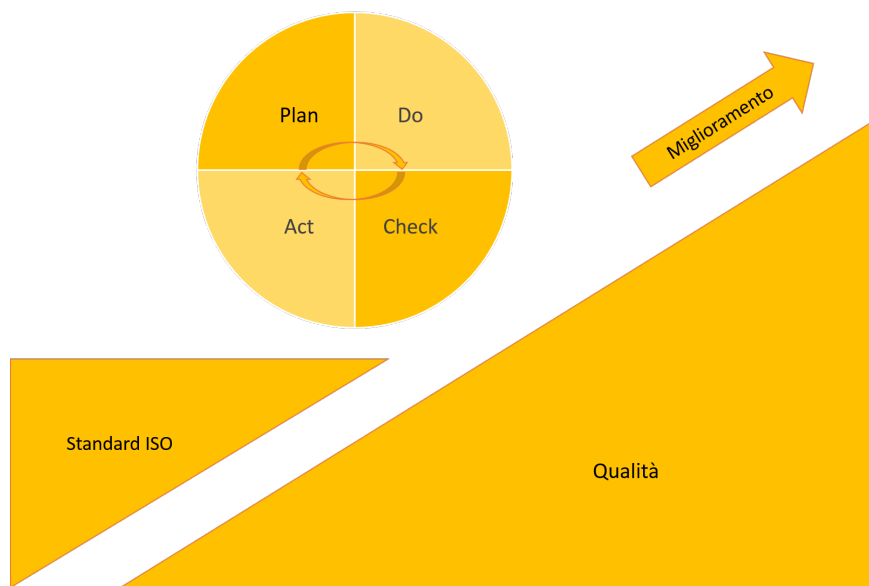


Figura 1: Ciclo di Deming

Il ciclo PDCA<sub>G</sub> è suddiviso in 4 fasi:

- **Plan:** fase di pianificazione che definisce:
  - Gli obiettivi da raggiungere;
  - Come svolgere le attività per conseguirli, tenendo in considerazione le risorse disponibili e le scadenze;
  - Le responsabilità di chi va ad eseguire le attività;
  - Le metriche per valutare i processi.
- **Do:** fase di implementazione ed esecuzione delle strategie progettate nella fase precedente;
- **Check:** fase di verifica nella quale vengono controllati i risultati della fase precedente (secondo le metriche fissate nella fase *Plan*) e in cui si confrontano i prodotti effettivi con quelli attesi;
- **Act:** fase che prevede il miglioramento dei processi: utilizzando i risultati della verifica, si progettano modifiche ai processi per migliorarli. Nello svolgimento di questa attività viene tenuto conto delle *best practice*<sub>G</sub>. Tutte le decisioni prese in questa fase vengono implementate a partire dalla successiva fase di *planning*.

Lo scopo di questo ciclo è il continuo incremento qualitativo del processo di sviluppo e, conseguentemente, del prodotto. L'appoggio agli *standard* e alle *best practice* permette di consolidare il livello raggiunto e di non retrocedere. Il miglioramento può progredire infinitamente nel tempo, data l'assenza di limiti fissati.

### 4.3 Standard ISO/IEC 9126

Lo standard ISO<sub>G</sub>/IEC<sub>G</sub> 9126 è stato creato con lo scopo di descrivere gli obiettivi qualitativi del prodotto e definire le metriche che possono misurare il raggiungimento di tali obiettivi. Vi sono infatti dei criteri suddivisi in tre aree:

- **Qualità in uso:** è la qualità del prodotto *software* visto dall'utilizzatore, che ne fa uso in uno specifico contesto;
- **Qualità esterna:** è la qualità del prodotto *software* visto dall'esterno, quando viene eseguito e testato in un ambiente di prova;
- **Qualità interna:** è la qualità del prodotto *software* visto dall'interno, facendo riferimento a caratteristiche implementative, come l'architettura o il codice che ne deriva.

Essendo inizialmente impossibile testare la qualità in uso, si è deciso di focalizzarsi sulla qualità interna ed esterna. Nello standard sono previste sei caratteristiche qualitative principali, suddivise a loro volta in sottocategorie che possono essere misurate quantitativamente:

- **Funzionalità:** capacità del prodotto di fornire le funzioni richieste
  - **Idoneità:** capacità del prodotto di fornire un insieme di funzioni appropriate per un'attività;
  - **Accuratezza:** capacità del prodotto di fornire risultati coerenti e corretti a seconda del grado di precisione richiesto;
  - **Interoperabilità:** capacità interattiva del prodotto *software* con uno o più sistemi;
  - **Sicurezza:** capacità del prodotto *software* di proteggere i dati e le informazioni, e a consentire solo a sistemi autorizzati di fare modifiche;
  - **Conformità funzionale:** capacità del prodotto *software* di essere aderente allo standard in materia di funzionalità.
- **Affidabilità:** capacità del prodotto di garantire un livello di prestazioni adeguato
  - **Maturità:** capacità del *software* di evitare fallimenti causati da errori;
  - **Tolleranza agli errori:** capacità del *software* di mantenere un adeguato livello di prestazioni nonostante errori o violazioni dell'interfaccia;
  - **Capacità di recupero:** capacità del *software* di ristabilire un livello di *performance* adeguato e di recuperare dati in caso di errori;
  - **Conformità di affidabilità:** capacità del prodotto *software* di essere aderente allo standard in materia di affidabilità.
- **Usabilità:** capacità del prodotto di essere facilmente comprensibile ed usabile, e di risultare interessante per l'utente
  - **Intelligibilità:** capacità del *software* di permettere all'utente di capire se il prodotto è adeguato e se possa essere utilizzato per dei compiti particolari;
  - **Apprendibilità:** capacità del *software* di consentire all'utente di imparare ad utilizzare le sue funzionalità;

- **Operabilità:** capacità del *software* di consentire ai suoi utenti di essere usato;
  - **Attrattività:** capacità del *software* di creare interesse negli utenti;
  - **Conformità di usabilità:** capacità del *software* di essere aderente allo standard in materia di usabilità.
- **Efficienza:** capacità del prodotto di fornire prestazioni adeguate a seconda delle risorse utilizzate
    - **Comportamento temporale:** capacità del *software* di dare una risposta in tempi di elaborazione adeguati, quando svolge le sue funzionalità;
    - **Utilizzo di risorse:** capacità del *software* di utilizzare le risorse adeguate per la sua esecuzione;
    - **Conformità di efficienza:** capacità del *software* di essere aderente allo standard in materia di efficienza.
- **Manutenibilità:** capacità del prodotto di essere modificato e aggiornato. Tali modifiche o aggiornamenti possono rispecchiare correzioni, miglioramenti o adattamenti del *software* nei confronti di cambiamenti ambientali, di specifiche o delle funzionalità
    - **Analizzabilità:** capacità del *software* di poter essere studiato per cercare difetti;
    - **Modificabilità:** capacità del *software* di consentire l'implementazione di una qualche modifica;
    - **Stabilità:** capacità del *software* di evitare errori o fallimenti causati da una o più modifiche;
    - **Testabilità:** capacità del *software* di permettere la validazione di una versione modificata del *software*;
    - **Conformità di manutenibilità:** capacità del prodotto *software* di essere aderente allo standard in materia di manutenibilità.
- **Portabilità:** capacità del *software* di poter essere spostato in vari ambienti di lavoro
    - **Adattabilità:** capacità del *software* di essere adattato a diversi ambienti senza necessitare di modifiche;
    - **Installabilità:** capacità del *software* di poter essere installato in ambienti differenti;
    - **Coesistenza:** capacità del *software* di coesistere con altri *software* indipendenti condividendo delle risorse;
    - **Sostituibilità:** capacità del *software* di poter sostituire un *software* simile che abbia le stesse funzionalità e che sia nello stesso ambiente;
    - **Conformità di portabilità:** capacità del prodotto *software* di essere aderente allo standard in materia di portabilità.

## 5 Resoconto delle attività di verifica

### 5.1 Riassunto delle attività di verifica

#### 5.1.1 Revisione dei Requisiti

Nel periodo antecedente alla consegna della documentazione sono state eseguite le attività di verifica dei processi e dei documenti redatti. Ogni documento è stato controllato tramite la tecnica del *walkthrough* descritta nella sezione 2.8.2 Walkthrough. Si sono riscontrati diversi errori che sono stati segnalati in modo informale ai redattori. Il gruppo è consapevole di non aver seguito la procedura descritta nella sezione 3.1 Comunicazione e risoluzione di anomalie. Pertanto si impegna a seguirla dalla prossima fase: Analisi di Dettaglio. Si precisa inoltre che nel corso di questa attività il gruppo non si è occupato di stilare la lista di controllo necessaria per la tecnica dell'*inspection*. Infine, dopo la correzione dei documenti, si è provveduto a calcolare le metriche descritte nella sezione 2.9 Misure e metriche.

Il tracciamento dei requisiti è avvenuto grazie all'utilizzo di un *software* realizzato appositamente dagli *Amministratori* del gruppo.

I processi sono stati verificati utilizzando le metriche di processo descritte nella sezione 2.8.1 Metriche per i processi. Sono quindi stati riportati i valori di BV e SV relativi a tutti i processi di ciascuna fase.

### 5.2 Dettaglio delle verifiche tramite analisi

#### 5.2.1 Documenti

Riportiamo in questa sezione i valori degli indici Gulpease calcolati per ogni documento. Un documento è valido se e solo se rispetta le metriche descritte nella sezione 2.9.2 Metriche per i documenti.

Documento	Valore	Indice
<i>Piano di Qualifica v1.0.0</i>	54	Superato
<i>Piano di Progetto v1.0.0</i>	58	Superato
<i>Norme di Progetto v1.0.0</i>	56	Superato
<i>Analisi dei Requisiti v1.0.0</i>	60	Superato
<i>Studio di Fattibilità v1.0.0</i>	54	Superato
<i>Glossario v1.0.0</i>	50	Superato

Tabella 1: Indici Gulpease dei documenti

La tabella mostra come i documenti rientrano tutti nel *range* ottimale precedentemente definito. Rispettano pertanto il livello di leggibilità minimo desiderato.

### 5.2.2 Processi

Qui vengono riportati i valori di SV e BV descritti nella sezione 2.8.1 Metriche per i processi, per le attività nella fase di **Analisi dei Requisiti** che hanno portato alla stesura dei documenti.

Attività	SV	BV
<i>Piano di Qualifica v1.0.0</i>	-30 €	-30 €
<i>Piano di Progetto v1.0.0</i>	0 €	-30 €
<i>Norme di Progetto v1.0.0</i>	0 €	-30 €
<i>Analisi dei Requisiti v1.0.0</i>	+50 €	+25 €
<i>Studio di Fattibilità v1.0.0</i>	+30 €	0 €
<i>Glossario v1.0.0</i>	0 €	0 €

Tabella 2: Metriche dei processi

Complessivamente la fase di **Analisi** ha:

- SV : +50€;
- BV : -65€.

Da questi indici si può notare che:

- Grazie a una pianificazione accurata delle scadenze, le attività sono state svolte nei tempi pianificati, alcune volte terminando in anticipo rispetto alle date di fine prefissate;
- Il costo complessivo si è di poco discostato da quanto pianificato nel *Piano di Progetto v1.0.0*. La causa di questa variazione dei costi è imputabile alla poca esperienza del gruppo nel redigere tale tipologia di documenti.

### 5.2.3 Esito delle revisioni

Durante lo sviluppo del progetto vi saranno quattro revisioni da parte del Committente. Il Committente segnalerà problematiche che verranno riscontrate valutando globalmente l'andamento del progetto e di ogni singolo documento. Grazie a queste informazioni sarà possibile correggere gli errori commessi, al fine di procedere lungo delle attività lavorative che risultino corrette e verificate.