Piano di Qualifica

Gruppo SWEetBIT — ***Progetto SWEDesigner***

Informazioni sul documento

Versione	1.0.0
Redazione	Sebastiano Bertolin
Verifica	Da inserire
Approvazione	Da inserire
Uso	Interno
Distribuzione	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Gruppo SWEetBIT

Descrizione

Questo documento descrive le strategie di verifica, adottate dal gruppo SWEet BIT, atte a garantire gli obiettivi qualitativi riguardanti il prodotto SWEDesigner.

Indice

Versioni del documento

Versione	Data	Persone coinvolte	Descrizione
0.0.1	AAAA/MM/GG	Sebastiano Bertolin	Stesura Introduzione e Difinizione Obbiettivi di Qualità

1 Introduzione

1.1 Scopo del documento

Questo documento descrive le strategie adottate dal gruppo SWEet BIT per il conseguimento degli obiettivi di qualità riguardanti il prodotto. Il raggiungimento di tali obiettivi è possibile solo tramite una precisa e continua verifica delle attività svolte; così facendo è possibile rilevare e correggere tempestivamente eventuali anomalie, minimizzando lo spreco di risorse.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazone di una *Web App*$_G$ che fornisca all'utente un *UML*$_G$ *Designer*$_G$ con il quale riuscire a disegnare correttamente diagrammi delle classi e descrivere il comportamento dei metodi interni alle stsse attraverso l'utilizzo di -da decidere il tipo di schema-. La *Web App*$_G$ permetterà all'utente di generare codice Java o Javascript dal diagramma disegnato ed eventualmente andare a ritoccarne il risultato al fine di ottenere un codice eseguibile, funzonante e funzionale.

1.3 Glossario

Con lo scopo di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v1.0.0*. In esso saranno definiti, in modo chiaro e conciso i termini che possono causare ambiguità o incomprensione del testo.

1.4 Riferimenti

1.4.1 Informativi

- **Analisi dei requisiti:** *Analisi dei requisiti v1.0.0*
- **Piano di progetto:** *Piano di progetto v1.0.0*
- **Lucidi dell'insegnamento di Ingegneria del *Software*:** http://www.math.unipd.it/~tullio/IS-1/2016/
- **Libro SWEBOK v3.0: *Chapter 10: Software Quality*:** https://www.computer.org/web/swebok/
- ***ISO*$_G$*/IEC*$_G$ 9126 *Software engineering — Product quality*:** https://en.wikipedia.org/wiki/ISO/IEC_9126

- ***ISO$_G$/IEC$_G$* 15504 *Information technology – Process assessment*:**
  https://en.wikipedia.org/wiki/ISO/IEC_15504

1.4.2 Normativi

- **Norme di progetto:** *Norme di progetto v1.0.0*
- **Capitolato di appalto SWEDesigner (C6):**
  http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf

2 Visione generale delle strategie di verifica

2.1 Definizione obiettivi di qualità

Questa sezione si presta a descrivere sia gli obiettivi di qualità, richiesti dal committente, riguardanti il prodotto che quelli relativi ai processi necessari alla sua produzione.

2.1.1 Qualità di prodotto

Il gruppo SWEet BIT si è impegnato ad aderire allo standard ISO_G/IEC_G 9126, in modo garantire le seguenti caratteristiche di qualità per il prodotto:

Funzionalità: il prodotto deve soddisfare i requisiti indicati nel documento *Analisi dei Requisiti*.

Affidabilità: il software prodotto deve essere robusto, deve essere prevista una completa gestione degli errori ed eccezioni, in modo da prevenire perdite di dati e facilitarne il ripristino.

Usabilità: il software prodotto deve risultare semplice e di facile utilizzo da parte dell'utente; esso deve essere di facile apprendimento, l'interfaccia utente deve risultare familiare e intuitiva. Il prodotto deve soddisfare al meglio le esigenze dell'utente finale.

Efficienza: Il software deve essere capace di fornire tutti i servizi attesi con il minimo dispendio di risorse.

Manutenibilità: Il software prodotto deve essere facilmente modificabile, in caso di correzione di errori e migliorie da apportare; esso deve essere facilmente adattabile in caso di cambio di ambiente operativo e/o cambio di requisiti.

Portabilità: Il software prodotto deve, per la parte *front end*$_G$, offrire gli stessi servizi nella maggior parte dei *browser*$_G$; la parte di *back end*$_G$ deve funzionare su vari sistemi operativi.

Per ognuna delle caratteristiche principali appena elencate esistono diverse sotto-caratteristiche, le quali sono approfondite nella sotto-sezione Standard e metodi per la gestione della Qualità dove è descritto lo standard ISO_G/IEC_G 9126

.

2.1.2 Qualità di processo

La qualità dei processi che portano allo sviluppo di un software hanno un ruolo fondamentale sulla qualità del prodotto. Il gruppo SWEet BIT ha deciso di adottare il

ciclo di Deming, o ciclo $PDCA_G$, come modello per il continuo miglioramento dei processi produttivi; come complemento al ciclo di Deming è stato scelto il modello descritto dallo standard ISO_G/IEC_G 15504 detto $SPICE_G$, il quale fornisce gli stumenti per la valutazione dei processi produttivi.

2.2 Organizzazione

Per onguno dei periodi descritto dal *Piano di progetto* è necessario seguire una verifica mirata al tipo di processo e relativo prodotto:

- **Analisi:** in questo periodo vengono controllati i processi e la documentazione prodotta, la verifica dovà essere eseguita secondo i metodi descritti nel documento *Norme di progetto*.
- **Analisi nel dettaglio:** durante questo periodo sono verificati i processi che portano all'incemento dei documenti, e i documenti stessi, prodotti durante il perido di analisi. Le procedure di verifica sono descritte nel documento *Norme di progetto*.
- **Progettazione architetturale:** in questo periodo vengono verificati i processi, e relativi prodotti, che portano all'incremento dei documenti redatti nel precedente periodo di analisi nel dettaglio; sempre in questa fase sono verificati i processi e prodotti riguardanti la progettazione architetturale. Le procedure di verifica sono descritte in dettaglio nel documento *Norme di progetto*.
- **Progettazione di dettaglio e codifica:** anche in questo periodo vengono verificati i processi, e relativi prodotti, che portano all'incremento dei documenti del periodo di progettazione architetturale. Vengono verificati processi e prodotti della sotto-fase di codifica. Le procedure di verifica sono descritte nel documento *Norme di progetto*.
- **Verifica e validazione:** sono verificati i processi che portano all'aggiornamento della documentazione e i documenti stessi. In questo periodo vengono effettuati i test per la validazione del prodotto finale.

2.3 Pianificazione strategica e temporale

Intendendo rispettare le scadenze specificate nel *Piano di progetto* è necessario un approccio organico e ben pianificato alle attività di verifica. Deve essere pianificato ogni incremento su codice e documentazione, di conseguenza lo dovranno essere tutte le attività di verifica in modo da avere il tempo di effettuare le dovute correzioni. Ogni processo di verifica dovrà essere automatizzato il più possibile, questo per poter, dove possibilie, risparmiare su risorse umane. L'utilizzo di software apposito, e i metodi, per la verifica sono descritti nel documento *Norme di progetto*.

2.4 Gestione delle responsabilità

Il *Responsabile di progetto* e i *Verificatori* hanno la responsabilità delle attività di verifica; i compiti di queste figure sono argomento trattato nel documento *Piano di progetto*.

2.5 Risorse

Si possono identificare due distinti tipi di risorse necessari a garantire la qualità dei prodotti e processi e sono:

- **Risorse umane:** sono distinte dai ruoli rivestiti dai componenti del gruppo SWEet BIT:
  - Responsabile
  - Amministratore
  - Analista
  - Progettista
  - Programmatore
  - Verificatore

  Ognuno dei sopracitati ruoli è descritto nel dettaglio nel documento *Piano di progetto*.
- **Risorse tecnologiche:** in questa categoria rientrano sia il software utilizzato per l'automazione delle verifiche, sia l'hardware neccessario per eseguire il suddetto software. Nel documento *Norme di progetto* sono descritti nel dettaglio tutti gli strumenti software utilizzati per verifica e validazione di processi e prodotti.

3 Strategia di verifica nel dettaglio

Questa sezione del documento descrive le metriche utilizzate per la quantificazione della qualità e le tecniche di analisi adottate.

3.1 Metriche software, metriche di processo e misurazioni

Le metriche ritenute necessarie per una corretta misurazione della qualità di processi e prodotti sono descritte in questa sezione; l'attività di verifica si baserà su queste metriche e per ognuna di queste è stato necessario stabilire due range:

- **Range di accettabilità:** l'intervallo di valori entro il quale deve trovarsi il risultato della misurazione di un processo, o prodotto, per essere ritenuto accettabile.
- **Range di ottimalità:** l'intervallo di valori in cui si deve trovare il risultato della misurazione di un processo, o prodotto, per essere considerato ottimale; questo range è l'obiettivo del gruppo SWEet BIT.

3.1.1 Metriche riguardanti i processi

Per la verifica dei processi sono state adottate due metriche di seguito descritte:

Schedule Variance (SV): È un indice che dà informazione necessarie a determinare se ci si trova in anticipo, in ritardo o in linea alle tempistiche delle attività di progetto. La seguente formula dà SV in termini di costo:

$$SV = BCWP - BCWS$$

Dove:

- BCWP = costo totale del lavoro svolto al momento della misurazione;
- BCWS = costo totale del lavoro pianificato al momento della misurazione.

SV ha tre significativi risultati:

- SV>0 indica che si è avanti rispetto alle pianificazione temporale del lavoro;
- SV=0 indica che si è in linea alle tempistiche delle attività di progetto;
- SV<0 indica che si è in ritardo rispetto alla pianificazione temporale delle attività.

I range stabiliti sono:

- Range di accettabilità = [≥-(*preventivo fase* * 5%)]
- Range di ottimalità = [≥0]

Cost Variance (CV): indica se vi sono state più o meno spese rispetto al previsto. La seguente formula dà CV in termini di costo:

$$CV = BCWP - ACWP$$

Dove:

- BCWP = costo totale del lavoro svolto al momento della misurazione;
- ACWP = costo totale richiesto per il completamento del lavoro al momento della misurazione.

CV ha tre significativi risultati:

- CV>0 indica che il progetto sta avendo un costo inferiore rispetto a quanto preventivato;
- CV=0 indica che il progetto ha un costo in linea a quanto preventivato;
- CV<0 indica che il progetto ha superato il costo preventivato.

I range stabiliti sono:

- Range di accettabilità = [≥-(*preventivo fase* * 10%)]
- Range di ottimalità = [≥0]

3.1.2 Metriche riguardanti i documenti

Indice Gulpease: è un idice di leggibilità di un testo per la lungua italiana. Questo indice considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere. La formula per il calcolo dell'indice Gulpease è:

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

Il risultato è un numero nell'intervallo [0-100], generalmente risulta che:

- inferiore a 80 il testo è difficile da leggere per chi ha la licenza elementare;
- inferiore a 60 il testo è difficile da leggere per chi ha la licenza media;
- inferiore a 40 il testo è difficile da leggere per chi ha un diploma superiore.

I range stabiliti sono:

- Range di accettabilità = [40-100]
- Range di ottimalità = [50-100]

3.1.3 Metriche riguardanti il prodotto software

Come descritto nello standard ISO_G/IEC_G 9126, si posso identificare due tipi di metriche utilizzate per misurare la qualita del prodotto software, esse sono:

- *Metriche interne*: vengono applicate al software non eseguibile durante le fasi di progettazione e codifica.
- *Metriche esterne*: servono per poter misurare il comportamento del prodotto software attraverso test effettuati in fase di esecuzione.

Per ogni caratteristica identificata nella §2.1.1 del presente documento, è stata trovata almeno una metrica interna e una esterna, esse sono:

- **Funzionalità**
  Come **metrica interna** viene utilizzata la **Completezza funzionale**; lo scopo è quello di poter quantificare il livello di copertura delle funzioni del software, sui requisiti individuati durante il periodo di analisi; la formula per il calcolo della completezza funzionale è la seguente:

  $$CompF = (Fs - Fo) / Ri$$

  Dove:
  - CompF = livello completezza funzionale;
  - Fs = numero di funzionalità sviluppate;
  - Fo = numero di funzionalità obbligatorie (che coprono *requisiti obbligatori*)
  - Ri = numero di *requisiti desiderabili.*

  I range stabiliti sono:
  - Range di accettabilità = [≥0.5]
  - Range di ottimalità = [≥1]

  Come **metrica esterna** si è deciso di utilizzare la **Correttezza funzionale**; questa metrica permette di misurare il livello di correttezza funzionale del software. La formula per il calcolo della correttezza funzionale è:

  $$CorrF = 1 - Fe / Fp$$

  Dove:
  - CorrF = livello correttezza funzionale;
  - Fe = numero funzioni che rilevate errate dai test;
  - Fp = numero di funzioni previste dalle specifiche.

  I range stabiliti sono:

  - Range di accettabilità = [0.75-1]
  - Range di ottimalità = [0.9-1]

- **Affidabilità**
  Come **metrica interna** si è optato per la **Rimozione errori**, la quale permette di misurare il livello di efficacia di rimozione degli errori; la formula per il calcolo della rimozione degli errori è:

  $$Re = Dc \ / \ Dr$$

  Dove:

  - Re = livello efficacia rimozione errori;
  - Dc = numero difetti corretti;
  - Dr = numero difetti rilevati.

  I range stabiliti sono:

  - Range di accettabilità = [0.8-1]
  - Range di ottimalità = [1]

  Come **metrica esterna** viene utilizzata la **Frequenza/Assenza di malfunzionamenti**, questa è applicata per misurare quanto efficaciemente sono gestite le situazioni in cui si verificano errori durante l'operatività del sistema. La formula è:

  $$FreqE = Mc \ / \ Mr$$

  Dove:

  - FreqE = indice di frequenza errori gestiti, e risolti, correttamente;
  - Mc = numero di malfunzionamenti gestiti e risolti correttamente;
  - Mr = numero di malfunzionamenti rilevati durante i test.

  I range stabiliti sono:

  - Range di accettabilità = [0.7-1]
  - Range di ottimalità = [0.95-1]

- **Usabilità Metrica interna** utilizzata è la **Completezza descrittiva**, la quale da un indice di completezza celle funzionalità descritte nella documentazione del prodotto. La formila per la misurazione è:

  $$CompDesc = Fd \ / \ Fp$$

  Dove:

  - CompDesc = indice di completezza descrittiva;

- Fd = numero di funzioni descritte nella documentazione del prodotto;
- Fp = numero di funzioni previste.

I range stabiliti sono:

- Range di accettabilità = [0.8-1]
- Range di ottimalità = [1]

Metrica esterna utilizzata è la **Efficacia documentazione**, che serve a misurare l'efficacia con la quale la documentazione viene compresa dall'utente, quando egli necessita di consultare la documentazione per comprendere una funzione. La formula per il calcolo è:

$$EffDoc = Fappr \ / \ Fnd$$

Dove:

- EffDoc = indice di efficacia documentazione;
- Fappr = numero di funzioni apprese e completate con successo dall'utente accedendo alla documentazione;
- Fnd = numero di funzioni totali in cui l'utente ha acceduto alla documentazione durante il test.

I range stabiliti sono:

- Range di accettabilità = [0.7-1]
- Range di ottimalità = [0.9-1]

3.2 Tecniche di analisi

3.2.1 Analisi statica

L'analisi statica è il processo di verifica del sistema o di un suo componente, senza che esso debba necessariamente poter essere eseguito. Questa tecnica permette di verificare sia la documentazione che il codice. Durante il processo di analisi statica vengono utilizzate le metriche di qualità interna per la verifica del prodotto software.

3.2.2 Analisi dinamica

L'analisi dinamica è il processo che verifica il prodotto software mentre esso è in esecuzione; viene effettuata tramite test che devono essere coerentemente pianificati in modo

da evitare spreco di risosrse. Durante l'attività di analisi dinamica vengono utilizzate le metriche di qualità esterna per la verifica del prodotto software in esecuzione. Organizzazione delle attività di test:

- **Test di unità**: L'obiettivo di questa fase è quello di riuscire a capire se ogni unità del codice si comporti esattamente come previsto, un unità di codice è la più piccola parte di software testabile.
- **Test di integrazione**: In questa fase di test viene testata una componente, una componente è l'aggragazione di più unità software. Questo tipo di test consente di trovare errori che si verificano nell'aggregazione di varie unità.
- **Test di sistema**: in questa fase viene verificato il comportamento del prodotto software finale, lo scopo è quello di capire se esso rispetta i requisiti individuati nella fase di analisi.
- **Test di regressione**: Questo test ha lo scopo di stabilire se le modifiche apportate al software hanno compromesso componenti software precedentemente funzionanti. Questi test vengono eseguiti ogni volta che viene apportata una modifica al software.

4 Gestione amministrativa della revisione

5 Pianificazione dei test

A Standard e metodi per la gestione della Qualità

A.0.1 *ISO_G/IEC_G 9126 Software engineering — Product quality*

L'obiettivo di questo standard è quello di definire un modello per poter valutar la qualità del software.
Lo standard si divide in quattro parti:

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

A.0.1.1 Modello della qualità del software Il modello della qualità del software presentato nella prima parte dello standard (ISO_G/IEC_G 9126-1), identifica sei caratteristiche generali e varie sottocaratteristiche:

1. **Funzionalità:**
   - **Adeguatezza**: la capacità del software di offrire un appropriato insieme di funzioni per determinati compiti.
   - **Accuratezza**: la capacità del prodotto software di rendere risultati concordati o i precisi effetti aspettati.
   - **Interoperabilità**: capacità del prodotto software di operare e interagire con sistemi uno o più sistemi specificati.
   - **Sicurezza**: la capacità di proteggere dati e informazioni, impedendo a persone e sistemi non autorizzati di accedervi, e garantendo sempre l'accesso ai dati a persone e sistemi autorizzati.
   - **Conformità funzionale**: capacità del prodotto software di aderire a standard, convenzioni e regolamentazioni riguardanti il settore operativo a cui vengono applicate.
2. **Affidabilità:**
   - **Maturità**: capacità del prodotto software di evitare il verificarsi di errori, malfunzionamenti o siano prodotto risultati errati.
   - **Tolleranza agli errori**: è la capacità del prodotto software di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto.

- **Recuperabilità**: capacità di ripristinare il livello appropriato di prestazioni e di recupero delle informazioni rilevanti, in seguito a un malfunzionamento. A seguito di un errore, il software può risultare non accessibile per un determinato periodo di tempo, questo arco di tempo è valutato proprio dalla caratteristica di recuperabilità.
- **Aderenza**: è la capacità di aderire a standard, regole e convenzioni inerenti all'affidabilità.

3. **Usabilità:**
   - **Comprensibilità**: è la capacità del prodotto software di mettere l'utente in grado di comprendere se il software è appropriato, e come esso possa essere usato per scopi e condizioni d'uso particolari
   - **Apprendibilità**: è la capacità di ridurre l'impegno richiesto agli utenti per imparare ad usare la sua applicazione.
   - **Operabilità**: è la capacità del prodotto software di rendere l'utente in grado di operarlo e controllarlo.
   - **Attrattiva**: è la capacità del software di essere piacevole per l'utente.
   - **Conformità**: è la capacità del prodotto software di aderire a standard o convenzioni relativi all'usabilità.
4. **Efficienza:**
   - **Comportamento rispetto al tempo**: è la capacità di fornire adeguati tempi di risposta, elaborazione e velocità di attraversamento, sotto condizioni determinate.
   - **Utilizzo delle risorse**: capacità del prodotto software di usare un adeguato quntitativo e tipo di risorse quando il software esegue le sue funzioni in determinate condizioni.
   - **Conformità**: è la capacità di aderire a standard e specifiche sull'efficienza.
5. **Manutenibilità:**
   - **Analizzabilità**: è la del prodotto software di essere facilmente controllato per la ricerca di errori, o di facilitare l'identificazione delle parti che devono essere modificate.
   - **Modificabilità**: capacità del prodotto software di rendere possibili eventuali implementazioni di modifiche.
   - **Stabilità**: la capacità del prodotto software di evitare effetti indesiderati dovuti alle modifiche del software stesso.

- **Testabilità**: è la capacità del prodotto software di poter validare le modifiche ad esso apportate
- **Conformità di manutenibilità**: è la capacità di aderire a standard e specifiche riguardanti la manutenibilità.

6. **Portabilità:**
   - **Adattabilità**: la capacità del software di essere adattato per differenti ambienti operativi senza dover applicare modifiche diverse da quelle fornite per il software considerato.
   - **Installabilità**: la capacità del software di essere installato in uno specificato ambiente.
   - **Sostituibilità**: è la capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti nello stesso ambiente.
   - **Conformità**: la capacità del prodotto software di aderire a standard e convenzioni relative alla portabilità.

A.0.1.2 Metriche per la qualità interna Le metriche per la qualità interna o mentriche interne, descritte nel *technical report* ISO_G/IEC_G 9126-3, sono delle metriche che si applicano al software non eseguibile durante le fasi di progettazione e codifica. Le metriche interne permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il software eseguibile Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale, poiché gli attributi interni influiscono su quelli esterni e quelli in uso.

A.0.1.3 Metriche per la qualità esterna Le metriche per la qualità esterna o metriche esterne, descritte nel *technical report* ISO_G/IEC_G 9126-2, sono delle metriche adatte alla misurazione dei comportamenti del software sulla base di misure ottenute da test, oparando e osservando il software eseguibile o sistema stesso.

A.0.1.4 Metriche per la qualità in uso Le metriche per la qualità in uso misurano il grado con cui il prodotto software permette agli utenti di svolgere le proprie attività con efficacia, produttività, sicurezza e soddisfazione nel contesto operativo previsto. Questa valutazione è quindi fatta in specifici scenari d'uso.
La qualità deriva anche dall'effetto combinato di più caratteristiche interne ed esterne di qualità. Il *technical report* ISO_G/IEC_G 9126-4 fornisce alcune metriche per la misurazione della qualità in uso.

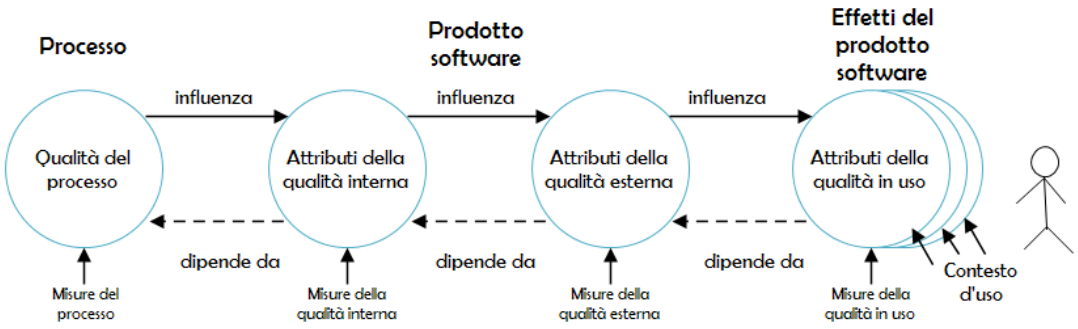

Figura 1: Ciclo di qualità del software

A.0.2 *ISO_G/IEC_G 15504 Information technology – Process assessment*

ISO_G/IEC_G 15504, anche conosciuta come SPICE (Software Process improvement and Capability Determination), è un insieme di nove documenti di standard tecnici relativi ai processi di sviluppo del software e relative funzioni di business e, in particolare, alla loro valutazione.

A.0.2.1 Modello di riferimento *ISO_G/IEC_G* 15504 presenta un modello di riferimento che definisce una dimensione del processo e una della capacità.
La dimensione del processo definisce processi divisi in cinque categorie di processo:

- *customer/suplier*;
- *engineering*;
- *supporting*;
- *management*;
- *organization*.

La dimensione della capacità definisce una scala di maturità a cinque livelli (più il livello base, detto livello 0) così definiti:

- *Level 5: Optimizing process*;
- *Level 4: Predictable process*;
- *Level 3: Established process*;
- *Level 2: Managed process*;
- *Level 1: Performed process*;
- *Level 0: Incomplete process*.

La capacità dei processi è misurata tramite gli attributi definiti a livello internazionale e che sono:

- *1.1 Process performance*;
- *2.1 Performance management*;
- *2.2 Work product management*;
- *3.1 Process definition*;
- *3.2 Process deployment*;
- *4.1 Process measurement*;
- *4.2 Process control*;
- *5.1 Process innovation*;
- *5.2 Process optimization*.

Ogni attributo del processo consiste di una o più pratiche generiche, le quali sono ulteriormente elaborate in "Indicatori della pratica" che aiutano nella fase di valutazione delle prestazioni.
Ciascun attributo del processo è valutato seconod una scala di quattro valori (N-P-L-F):

- *Not achieved* (0 - 15%);
- *Partially achieved* (>15% - 50%);
- *Largely achieved* (>50% - 85%);
- *Fully achieved* (>85% - 100%).

La valutazione si basa su prove raccolte fronte di ciascun indicatore di processo durante la fase di valutazione.

A.0.2.2 Valutazione ISO_G/IEC_G 15504 fornisce una guida per effettuare una verifica dei processi:

- il processo di valutazione (*the assessment process*);
- il modello di valutazione (*the model for the assessment*);
- tutti strumenti per effettuare la valutazione (*any tools used in the assessment*).

A.0.2.3 Processo di valutazione Eseguire la valutazione dei processi è oggetto della parte 2 e 3 del ISO_G/IEC_G 15504.
Il processo di valutazione può essere generalizzato nei seguenti passi:

- inizio dell'assessment;

- selezione del valutatore e del team di valutazione;
- pianificazione dell'assessment, inclusa la definizione dei processi e dell'organizzazione da valutare;
- riunione preliminare;
- raccolta dei dati;
- validazione dei dati raccolti;
- valutazione del processo;
- rapporto sul risultato della valutazione.

A.0.3 Ciclo di Deming o ciclo di Shewhart

Il ciclo di Deming (o ciclo di Shewhard) detto anche PDCA (o PDSA) si compone di 4 fasi:

- **P - Plan**: questa fase stabilisce gli obiettivi e processi necessari all'ottenimento dei risultati coerenti ai risultati attesi; lo stabilire le aspettative di risultati, la completezza e accuratezza delle specifice scelte sono parte del miglioramento mirato. Iniziare su piccola scala, quando possibile, per poter verificare possibili effetti.
- **D - Do**: Implementazione della fase *Plan*, esecuzione del processo, creazione del prodotto. Raccogliere i dati per la creazione di grafici e analisi da destinare alla fase di "Check" e "Act".
- **C - Check(or S - Study)**: Controllo e studio dei risultati raccolti nella fase *Do*. Confrontare i risultati con quelli attesi, stabiliti nella fase *Plan*.
- **A - Act**: Azione per rendere definitivo e/o migliorare il processo. L'informazione ottenuta nella fase *Check* è utile per poter individuare i possibili punti di un processo dove apportare modifiche.

B Resoconto delle attività di verifica

B.1 Revisione dei requisiti

B.2 Revisione di progettazione

B.3 Dettaglio delle verifiche tramite analisi

B.3.1 Analisi

B.3.2 Processi

B.3.3 Documenti

B.4 Dettaglio esito delle revisioni