



Piano di Qualifica

Gruppo DigitalCookies — Progetto SWEDesigner

digitalcookies.group@gmail.com

Informazioni sul documento

Versione	1.0.0
Redazione	Carlo Sindico, Alberto Giudice, Alessia Bragagnolo
Verifica	Christian Cabrera
Approvazione	Alberto Rossetti
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo DigitalCookies

Descrizione

Questo documento descrive le operazioni di verifica e validazione seguite dal gruppo DigitalCookies relativi al progetto SWEDesigner.

Registro delle modifiche

Versione	Data	Collaboratori	Ruolo	Descrizione
1.0.0	23-03-2017	Alberto Rossetti	Responsabile	Approvazione
0.1.0	22-03-2017	Christian Cabrera	Verificatore	Verifica del documento
0.0.7	21-03-2017	Alberto Giudice	Analista	Stesura sezione Resoconto delle attività di verifica
0.0.6	20-03-2017	Alessia Bragagnolo	Analista	Stesura sezione Pianificazione ed esecuzione del collaudo e appendice B
0.0.5	16-03-2017	Alberto Giudice	Verificatore	Stesura sezione Gestione amministrativa della revisione
0.0.4	14-03-2017	Carlo Sindico	Verificatore	Stesura sezione Misure e Metriche
0.0.3	09-03-2017	Alberto Giudice	Verificatore	Stesura sezione Visione generale della strategia di verifica
0.0.2	08-03-2017	Carlo Sindico	Verificatore	Stesura sezione Introduzione
0.0.1	08-03-2017	Alberto Rossetti	Responsabile	Creazione del template

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Ambiguità	5
1.4	Riferimenti	5
1.4.1	Normativi	5
1.4.2	Informativi	5
2	Visione generale della strategia di verifica	6
2.1	Procedure di controllo di qualità di processo	6
2.2	Procedure di controllo di qualità di prodotto	6
2.3	Organizzazione	7
2.3.1	Analisi dei Requisiti di Massima	7
2.3.2	Analisi dei Requisiti di Dettaglio	7
2.3.3	Progettazione Architettuale	7
2.3.4	Progettazione di Dettaglio	7
2.3.5	Codifica	7
2.3.6	Validazione	8
2.4	Strategia	8
2.5	Responsabilità	8
2.6	Risorse	8
3	Misure e Metriche	10
3.1	Metriche per i documenti	10
3.1.1	Gulpease	10
3.2	Metriche per i processi	11
3.2.1	Schedule Variance	11
3.2.2	Budget Variance	12
3.3	Metriche per il software	12
3.3.1	Complessità ciclomatica	12
3.3.2	Numero di metodi - NOM	13
3.3.3	Bugs per line of code	13
3.3.4	Variabili non utilizzate e non definite	13
3.3.5	Numero parametri per metodo	14
3.3.6	Halstead	14
3.3.7	Halstead difficulty per-function	15
3.3.8	Halstead volume per-function	15
3.3.9	Halstead effort per-function	15
3.3.10	Mantenibility index	16
3.4	Riepilogo	16

4	Gestione amministrativa della revisione	17
4.1	Comunicazione delle anomalie	17
4.2	Procedure di controllo per la qualità di processo	17
5	Pianificazione ed esecuzione del collaudo	19
A	Resoconto delle attività di verifica	20
A.1	Riassunto delle attività di verifica	20
A.1.1	Revisione dei requisiti	20
A.2	Dettaglio delle verifiche tramite analisi	20
A.2.1	Analisi dei Requisiti di Massima	20
A.2.1.1	Documenti	20
A.2.1.2	Processi	20
B	Qualità	22
B.1	Ciclo di Deming	22
B.2	Qualità dei processi	23
B.3	Qualità del prodotto software	23

Elenco delle tabelle

1	Riepilogo delle metriche e dei range di accettazione e ottimali	16
2	Esiti del calcolo dell'indice Gulpease - <i>Analisi dei Requisiti di Massima</i> . .	20
3	Esiti del calcolo degli indici Schedule Variance e Budget Variance - <i>Analisi dei Requisiti di Massima</i>	21

Elenco delle figure

1	Il ciclo di miglioramento dei processi	18
2	Continuous quality improvement with PDCA	22

1 Introduzione

L'obiettivo primario da perseguire è la *qualità_G* del *prodotto_G* e dei suoi processi, ottenibile tramite una serie di controlli stabiliti inizialmente. L'assenza di queste verifiche, abbinata ad un *team di sviluppo_G* con più componenti senza particolari accortezze e competenze, portano al progressivo deterioramento del materiale prodotto, sia esso sorgente o documentazione.

Bisogna pertanto prevenire l'inserimento di materiale non conforme alle *Norme di Progetto v1.0.0* in quanto si avvierebbe un graduale degrado della sua qualità.

1.1 Scopo del documento

Il Piano di Qualifica illustra la strategia di *verifica_G* e *validazione_G* che il gruppo DigitalCookies ha deciso di adottare. È necessario dare una dimensione alla qualità dei prodotti e dei processi, operazioni che non rientrano nei normali ruoli di progetto, bensì rappresentano una *funzione aziendale_G*. Secondo le strategie riportate in questo documento il *committente_G* sarà in grado di valutare oggettivamente quanto è stato prodotto e disporrà di una solida base di verifica.

1.2 Scopo del prodotto

Lo scopo del prodotto è creare un software di costruzione di diagrammi *UML_G* con relativa generazione di codice *Java_G*. Il codice potrà essere generato dall'utente a partire dai diagrammi UML delle *classi_G* e da una versione modificata del diagramma delle *attività_G*.

L'utente, interagendo con il sistema, sarà in grado di:

- delineare la struttura delle classi utilizzando lo standard UML;
- definire il corpo dei metodi delle classi sfruttando una versione modificata del diagramma delle attività;
- generare un applicativo scritto in codice Java a partire dai diagrammi sopracitati.

L'utente potrà inoltre sfruttare la *libreria_G* fornita con il prodotto per generare con facilità diagrammi relativi al dominio dei giochi di carte.

L'*editor_G* sarà fruibile dall'utente attraverso un *browser_G* desktop idoneo all'utilizzo delle tecnologie *HTML5_G*, *CSS3_G* e *JavaScript_G*.

1.3 Ambiguità

Al fine di evitare ogni ambiguità relativa al linguaggio impiegato nei documenti viene fornito il *Glossario v1.0.0*, contenente la definizione dei termini in corsivo marcati con una G pedice.

1.4 Riferimenti

1.4.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0*;
- **Verbale di incontro esterno** con il *proponente_G* Zucchetti S.p.A. del 23-02-2017;
- ***Capitolato_G* d'appalto C6:** *SWEDesigner*: editor di diagrammi UML con generazione di codice
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf>;
- **Standard ISO/IEC 15504:** http://en.wikipedia.org/wiki/ISO/IEC_15504;
- **Standard ISO/IEC 9126:** http://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Standard IEEE 610.12-90:** https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=2677.

1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto v1.0.0*;
- **Slide del corso di Ingegneria del Software:** Qualità del software <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L10.pdf>;
- **SWEBOK 2004 Version - capitolo 11:** <http://www.computer.org/portal/web/swebok/htmlformat>;
- **Slide del corso di Ingegneria del Software:** Qualità di *processo_G* <http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L11.pdf>;
- **Ciclo di Deming:** https://it.wikipedia.org/wiki/Ciclo_di_Deming;
- **Indice *Gulpease_G*:** https://it.wikipedia.org/wiki/Indice_Gulpease.

2 Visione generale della strategia di verifica

La strategia generale adottata è quella di automatizzare il più possibile il lavoro di verifica; questo richiede scelta e uso di *tools_G* adeguatamente configurati. L'obiettivo è avere un riscontro affidabile e numericamente trattabile che permetta di assicurare il grado di qualità predeterminato. L'aspettativa è la riduzione del lavoro manuale permettendo così un'attività di verifica più semplice.

2.1 Procedure di controllo di qualità di processo

Le linee guida per la gestione della qualità seguono il modello *PDCA_G* (si veda appendice B) e descrivono come devono essere attuate le procedure di controllo:

- Pianificazione dettagliata;
- Monitoraggio delle attività pianificate;
- Definizione delle risorse necessarie al conseguimento degli obiettivi;
- Utilizzo di metriche per verificare il miglioramento della qualità dei processi.

Nel *Piano di Progetto v1.0.0* è presente una descrizione dettagliata della pianificazione di tali attività per il miglioramento continuo dei processi, che è ottenibile indirettamente con una costante analisi della qualità di prodotto. Un prodotto di bassa qualità è indice del fatto che a monte è presente un processo migliorabile.

Inoltre, al fine di quantificare la qualità dei processi, sono state adottate delle metriche, descritte nella sezione 3.

2.2 Procedure di controllo di qualità di prodotto

Il controllo di qualità del prodotto sarà garantito grazie ai seguenti processi:

- **Software Quality Assurance (*SQA_G*)**: l'insieme delle attività volte a garantire il raggiungimento degli obiettivi di qualità, in un processo che deve essere preventivo e non correttivo;
- **Verifica**: assicura che non vengano introdotti errori nel prodotto dalle attività dei processi svolti. Per l'intera durata del progetto verranno svolte attività di verifica sugli output dei processi, volte a garantire che essi siano corretti, completi e che rispettino le regole, le convenzioni e le procedure. I risultati di tali processi sono riportati nell'appendice A;
- **Validazione**: la conferma oggettiva che il prodotto finale soddisfi i requisiti e le aspettative.

2.3 Organizzazione

Viene verificata la qualità dei singoli processi e dei loro output. La verifica degli output dei periodi descritti nel *Piano di Progetto v1.0.0* prevede attività di verifica mirate, grazie anche alla possibilità di esaminare i soli cambiamenti incrementali esaminando i diari delle modifiche.

2.3.1 Analisi dei Requisiti di Massima

Durante tale attività verrà verificata la conformità dei processi e della documentazione prodotta rispetto le *Norme di Progetto v1.0.0*. Sarà inoltre verificata la corrispondenza tra requisiti e casi d'uso.

2.3.2 Analisi dei Requisiti di Dettaglio

Durante tale attività verranno ampliati i requisiti richiesti dal sistema e migliorati i documenti attuando delle correzioni in base all'esito della Revisione dei Requisiti. Tali modifiche saranno verificate per consistenza e conformità.

2.3.3 Progettazione Architetture

Durante tale attività avverrà la verifica dei processi incrementali relativi all'analisi e ai nuovi documenti di progettazione. Viene inoltre verificato che i test siano adeguatamente pianificati, secondo le tempistiche descritte nel *Piano di Progetto v1.0.0*, ed eseguiti secondo quanto descritto nelle *Norme di Progetto v1.0.0*.

2.3.4 Progettazione di Dettaglio

Durante tale attività è prevista la stesura, in modo dettagliato, dell'intero sistema, specificando in modo approfondito il comportamento e l'iterazione tra i vari componenti. Ogni requisito dovrà essere tracciabile nelle componenti individuate in questa attività.

2.3.5 Codifica

Durante tale attività sarà compiuta la verifica delle attività di codifica tramite tecniche di analisi statica e dinamica, come descritto nelle *Norme di Progetto v1.0.0*.

2.3.6 Validazione

Durante tale attività vengono effettuati tutti i test necessari per garantire che il prodotto soddisfi tutti i requisiti e funzioni correttamente.

Il *Registro delle Modifiche* viene incluso in ogni documento, in modo da mantenere uno storico delle modifiche apportate e delle relative responsabilità.

2.4 Strategia

Il *Piano di Progetto v1.0.0* fissa una serie di scadenze improrogabili, per rispettare le quali risulta necessario definire con chiarezza una strategia di qualifica efficace e organizzata.

Gli incrementi della documentazione e del codice possono essere di natura programmata, quindi prefissati a calendario, oppure possono insorgere come inaspettati. Per evitare quest'ultima eventualità ciascuna attività di redazione o codifica sarà preceduta da uno studio preventivo, in modo da ridurre la possibilità di commettere errori ed imprecisioni di natura tecnica e/o concettuale. Nell'eventualità dovessero comunque risultare necessario verranno programmate le dovute modifiche, come nel caso di *bug_G* o errori (vedi sezione 4.1).

La qualità di ogni incremento è basata sul fatto che la struttura di qualifica garantisce il rispetto delle *Norme di Progetto v1.0.0*, controllo svolto con l'aiuto di automatismi che segnaleranno le problematiche rilevate in modo da permetterne una rapida correzione, senza un eccessivo utilizzo di risorse umane. L'implementazione di tali controlli viene descritta nelle *Norme di Progetto v1.0.0*.

2.5 Responsabilità

La responsabilità della verifica viene attribuita al *Responsabile* di Progetto e ai *Verificatori*. I compiti e le modalità di attuazione sono definiti nel *Piano di Progetto v1.0.0*.

2.6 Risorse

La qualifica dei processi, essendo essa stessa un processo, consuma due tipologie di risorse:

- **Umane:** le figure coinvolte sono il *Responsabile* di Progetto e il *Verificatore_G*, i cui processi effettuati consumano ore di produttività contabilizzate e schedate secondo il *Piano di Progetto v1.0.0*, che ne definisce anche l'aspetto economico;



- **Tecnologiche:** riguardano i mezzi adottati per utilizzare degli automatismi per la qualità. Trattandosi esclusivamente di mezzi informatici il consumo si limita a *unità_G* di calcolo considerate a costo nullo, dato che tutte le elaborazioni sono svolte su mezzi per i quali non è richiesto né un contributo economico, né un quantitativo temporale degno di nota. Le modalità in cui tali risorse vengono impiegate sono descritte all'interno del documento *Norme di Progetto v1.0.0*.

3 Misure e Metriche

Vengono adottate delle metriche per rendere misurabili e valutabili i processi, i documenti e il software prodotto. Non c'è nessuna comparazione tra di essi, la visione serve al gruppo per tener monitorati l'andamento dei processi e la qualità del prodotto.

3.1 Metriche per i documenti

La leggibilità dei documenti è indispensabile per garantirne la qualità. Il gruppo ha scelto di utilizzare un indice per misurare la leggibilità dei testi in lingua italiana.

3.1.1 Gulpease

L'indice Gulpease è un indice di leggibilità di un testo tarato sulla lingua italiana. Rispetto ad altri ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico e permettendo di misurare la complessità dello stile di un documento. 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:

$$89 + \frac{300 \times (\text{numero delle frasi}) - 10 \times (\text{numero delle lettere})}{\text{numero delle parole}} \quad (1)$$

Il risultato è un valore compreso nell'intervallo tra 0 e 100, dove il valore 100 indica la più alta leggibilità, mentre 0 la più bassa. In generale risulta che testi con indice:

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

Inoltre non devono essere sottovalutate le seguenti considerazioni su tale indice:

- non indica la comprensibilità del testo, il contenuto delle cui frasi potrebbe essere totalmente non comprensibile, ma avere lo stesso un ottimo indice Gulpease;
- per lo scopo dei documenti saranno spesso usati termini tecnici che non si possono sostituire;
- interrompere frasi a favore di un indice più alto potrebbe storpiarne il contenuto;
- usare frasi troppo dirette potrebbe risultare poco professionale. Per queste ragioni i documenti saranno sempre valutati da un membro del gruppo, per stabilire se e come il testo possa essere semplificato.

Parametri utilizzati

- **Range di accettazione:** [40-100];
- **Range ottimale:** [60-100].

Per calcolare tale indice in tutti i documenti è stato utilizzato uno strumento citato nelle *Norme di Progetto v1.0.0*.

3.2 Metriche per i processi

Le seguenti metriche rappresentano un indicatore utilizzato per monitorare e prevedere l'andamento delle principali variabili critiche del progetto: tempi e costi. Sono utilizzate metriche di tipo *consuntivo_G* perché danno un riscontro immediato sullo stato attuale del progetto. Ad ogni incremento verranno valutati tali indici e, se necessario, verranno stabiliti opportuni provvedimenti da parte del *Responsabile* di Progetto.

3.2.1 Schedule Variance

Permette di calcolare le tempistiche raggiunte alla data corrente rispetto alla schedulazione delle attività pianificate. È un indicatore di efficacia importante per il cliente.

$$SV = BCWP - BCWS$$

Dove:

- **BCWP:** indica il valore delle attività realizzate alla data corrente;
- **BCWS:** indica il costo pianificato per realizzare le attività di progetto alla data corrente.

Quindi con:

- **SV>0:** il lavoro prodotto è in anticipo rispetto quanto pianificato;
- **SV<0:** il lavoro è in ritardo;
- **SV=0:** il lavoro è in linea con quanto stabilito.

Parametri utilizzati

- **Range di accettazione:** $\geq -5\%$;
- **Range ottimale:** ≥ 0 .

3.2.2 Budget Variance

Permette di raffrontare i costi sostenuti alla data corrente rispetto al budget preventivato. È un indicatore finanziario.

$$BV = BCWS - ACWP$$

Dove:

- **BCWS**: indica il costo pianificato per realizzare le attività di progetto alla data corrente;
- **ACWP**: indica il costo effettivamente sostenuto per realizzare le attività di progetto alla data corrente.

Quindi:

- **BV > 0**: il budget sta venendo speso più lentamente di quanto pianificato;
- **BV < 0**: il budget sta venendo speso più velocemente di quanto pianificato;
- **BV = 0**: il budget speso è in linea con quanto stabilito.

Parametri utilizzati

- **Range di accettazione**: $\geq -10\%$;
- **Range ottimale**: ≥ 0 .

3.3 Metriche per il software

3.3.1 Complessità ciclomatica

La complessità ciclomatica è una metrica software che indica la complessità di un programma misurando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso. In tale grafo i *nod*i rappresentano unità atomiche di istruzioni, mentre gli *archi* connettono due nodi se le istruzioni ad esse associati possono essere eseguite in modo consecutivo.

Questo indice può essere applicato indistintamente a singole funzioni, moduli, metodi o *packages_G* di un programma e viene utilizzato per limitare la complessità durante le attività di sviluppo. Durante l'identificazione dei test è inoltre utile per determinare il numero di casi di test necessari, infatti l'indice di complessità ciclomatica è un limite superiore al numero di test necessari per raggiungere il *coverage* completo del modulo testato.

Parametri utilizzati

- **Range-accettazione**: $[0-25]$;

- **Range-ottimale:** $[0-10]^1$.

3.3.2 Numero di metodi - NOM

Il *number of methods* è una metrica usata per calcolare la media delle occorrenze di metodi per package. Un package non dovrebbe contenere un numero eccessivo di metodi, per cui valori superiori al range ottimale massimo potrebbero indicare la necessità di maggiore decomposizione del package.

Parametri utilizzati:

- **Range-accettazione:** $[3-10]$;
- **Range-ottimale:** $[3-7]$.

3.3.3 Bugs per line of code

Questa metrica misura il numero di bugs trovati su un certo quantitativo di linee di codice. All'aumentare del codice prodotto ci sarà anche un incremento della probabilità di errori nascosti. Un'ottima contromisura a questo problema è rendere il codice più chiaro e più semplice possibile. Con la crescita del codice prodotto è utile monitorare il rapporto tra i difetti trovati e il codice in fase di incremento, tale indice dovrebbe rimanere costante o diminuire nel tempo. In uno studio² di Steve McConnell, un esperto di software engineering, è stato provato che per un'azienda di medie dimensioni la quantità di bug per mille linee di codice varia in un range di $[15-50]$. **Parametri utilizzati:**

- **Range-accettazione:** $[0-60]$;
- **Range-ottimale:** $[0-20]$.

3.3.4 Variabili non utilizzate e non definite

La presenza di variabili non utilizzate viene considerata *pollution_G* e pertanto non viene accettata. Per sua natura JavaScript non blocca l'insorgenza di tali occorrenze, che verranno rilevate analizzando l'*Abstract syntax tree_G* (AST) ed eseguendo una cross-reference tra le variabili dichiarate e quelle iniziate.

Parametri utilizzati:

- **Range-accettazione:** $[0-0]$;

¹McCabe (dicembre 1976). A Complexity Measure. IEEE Transactions on Software Engineering: 308-320.

²Steve McConnell. Code Complete: A Practical Handbook of Software Construction, Second Edition., 20: 474-475.

- **Range-ottimale:** $[0-0]$.

3.3.5 Numero parametri per metodo

Indica il numero di parametri formali forniti in input a un metodo. Un valore elevato potrebbe indicare un metodo troppo complesso.

Parametri utilizzati:

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

3.3.6 Halstead

La metrica di $Halstead_G$ è un indice di complessità che identifica le proprietà misurabili del software e le relative relazioni. Si basa sull'osservazione che una metrica dovrebbe valutare l'implementazione di un algoritmo in linguaggi differenti ed essere indipendente dall'esecuzione su una specifica piattaforma. In un problema vengono identificati:

- n_1 = il numero di operatori distinti;
- n_2 = il numero di operandi distinti;
- N_1 = il numero totale di operatori;
- N_2 = il numero totale di operandi.

Da cui vengono calcolati:

- $n=n_1+n_2$: vocabolario della funzione;
- $N=N_1+N_2$: lunghezza della funzione.

Data la bassa disponibilità in rete di valori di riferimento, i range specificati sono frutto di un confronto tra il $report_G$ sulla complessità di una libreria *open source* presa come esempio (<https://github.com/philbooth/complexity-report/blob/master/EXAMPLE.md>) e i valori dichiarati in (<http://www.mccabe.com/pdf/McCabe20IQ20Metrics.pdf>). Tali valori vengono dichiarati momentanei (RR) e saranno da rivalutare sia considerando altre fonti, sia considerando i valori rilevanti in parti del codice che il gruppo considera come riferimento. Per calcolare gli indici di Halstead verrà utilizzato lo strumento JSMeter citato nelle *Norme di Progetto v1.0.0*.

3.3.7 Halstead difficulty per-function

Il livello di difficoltà di una funzione misura la propensione all'errore ed è proporzionale al numero di operatori presenti.

$$D = \left(\frac{n_1}{2}\right) * \left(\frac{N_2}{n_2}\right) \quad (2)$$

Parametri utilizzati:

- **Range-accettazione:** [0-30];
- **Range-ottimale:** [0-15].

3.3.8 Halstead volume per-function

Il volume descrive la dimensione dell'implementazione di un algoritmo e si basa sul numero di operazioni eseguite e sugli operandi di una funzione. Il volume di una function senza parametri composta da una sola linea è 20, mentre un indice superiore a 1000 indica che probabilmente la funzione esegue troppe operazioni.

$$V = N * \log_2 n \quad (3)$$

Parametri utilizzati:

- **Range-accettazione:** [20-1500];
- **Range-ottimale:** [20-1000].

3.3.9 Halstead effort per-function

Lo sforzo per implementare o comprendere il significato di una funzione è proporzionale al volume e al suo livello di difficoltà.

$$E = V * D \quad (4)$$

Parametri utilizzati:

- **Range-accettazione:** [0-400];
- **Range-ottimale:** [0-300].

3.3.10 Mantenibility index

Questa metrica³ è una scala logaritmica da $-\infty$ a 171 e viene calcolata sulla base delle linee di codice logiche, della complessità ciclomatica e dell'indice Halstead effort. Un valore alto indica una maggiore *manutenibilità*_G.

Parametri utilizzati:

- **Range-accettazione:** [70-171];
- **Range-ottimale:** [90-171].

3.4 Riepilogo

Metriche	Range accettazione	Range ottimale
Indice Gulpease	40-100	60-100
Schedule Variance	$\geq -5\%$	≥ 0
Budget Variance	$\geq -10\%$	≥ 0
Complessità ciclomatica	0-25	0-10
Numero di Metodi - NOM	3-10	3-7
Bugs per line of code	0-60	0-20
Variabili non utilizzare e non definite	0	0
Numero parametri per metodo	0-8	0-4
Halstead difficulty per-function	0-30	0-15
Halstead volume per-function	20-1500	20-1000
Halstead effort per-function	0-30	0-15
Mantenibility index	70-171	90-171

Tabella 1: Riepilogo delle metriche e dei range di accettazione e ottimali

³Definita nel 1991 da Paul Oman e Jack Hagemester alla University of Idaho.

4 Gestione amministrativa della revisione

4.1 Comunicazione delle anomalie

Il processo di *Software Quality Management_G* è finalizzato alla ricerca dei difetti. L'identificazione delle anomalie ne permette la correzione e informa il *Responsabile* di Progetto sullo stato del prodotto. Distinguere e catalogare le anomalie è utile per discutere, durante le revisioni e le riunioni, su quali correzioni attuare e con quale priorità. Di seguito vengono elencate le definizioni di anomalie (IEEE 610.12-90) adottate dal gruppo:

- **Error**: differenza riscontrata tra risultato di una computazione e valore teorico atteso (e.g. uscita dal range di accettazione degli indici di misurazione);
- **Fault**: un passo, un processo o un dato definito in modo erraneo (e.g. violazioni di norme tipografiche da parte di un documento), che corrisponde a quanto viene definito come bug;
- **Failure**: il risultato di un *fault* (e.g. incongruenza del prodotto con funzionalità indicate nell'*Analisi dei Requisiti v1.0.0*, incongruenza del codice con la definizione del prodotto);
- **Mistake**: azione umana che produce un risultato errato (e.g. anomalie nel *repository_G*).

La distinzione tra i tipi di anomalie consente di impostare le metriche per valutarne l'andamento e in alcuni casi predirlo; in particolare è stata scelta la metrica che conteggia il numero di *Bugs per lines of code*.

4.2 Procedure di controllo per la qualità di processo

Le procedure di controllo per la qualità di processo servono a migliorare la qualità del prodotto e/o diminuire i costi e i tempi di sviluppo. I principali approcci possibili sono due:

- **A maturità di processo**: riflette le buone pratiche di *management* e tecniche di sviluppo, ponendo come obiettivo primario la qualità del prodotto e la prevedibilità dei processi;
- **Agile**: sviluppo iterativo senza l'*overhead* della documentazione e di tutti gli aspetti predeterminabili. Viene caratterizzato dalla responsività ai cambiamenti dei requisiti del cliente e da uno sviluppo rapido.

Il team adotterà il primo approccio, ritenuto più adatto ad un gruppo inesperto. Con una visione proattiva il team cercherà di avere maggior controllo e previsione sulle attività da svolgere, approccio anche indicato come *best practice_G* per gruppi poco esperti. Il processo con maggiore influenza sulla qualità del sistema non è quello di sviluppo ma quello di progettazione. In tale processo le capacità e le esperienze dei singoli danno un

contributo decisivo. Il miglioramento dei processi è un processo ciclico composto da tre sotto-processi:

- **Misurazione:** misura gli attributi del progetto e punta ad allineare gli obiettivi con le misurazioni effettuate. Si forma così una *baseline_G* che aiuta a capire l'effettiva efficacia dei miglioramenti;
- **Analisi:** vengono identificate le problematiche e i colli di bottiglia dei processi;
- **Modifica:** sulla base delle problematiche riscontrate vengono proposti dei cambiamenti.

Il gruppo procederà nel seguente modo:

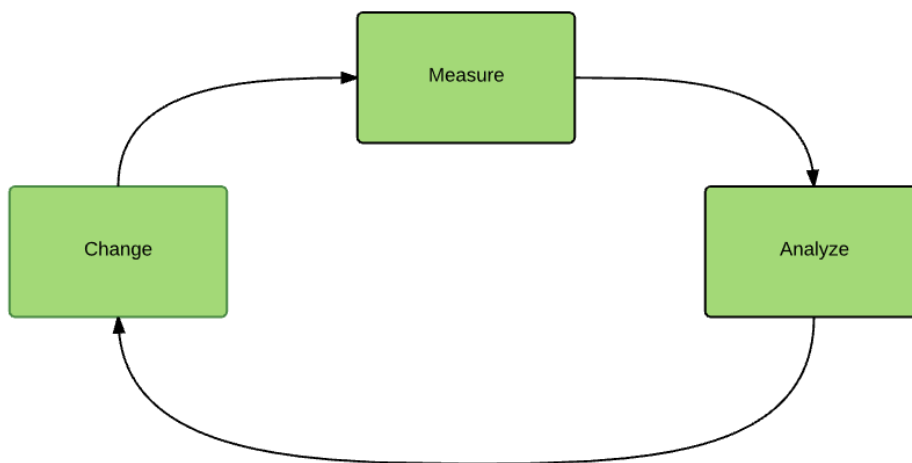


Figura 1: Il ciclo di miglioramento dei processi

- nella sezione *Dettaglio delle verifiche tramite analisi* (A.2) verranno inserite le misurazioni rilevate sulle metriche descritte in *Misure e Metriche* (3);
- l'analisi viene effettuata i giorni precedenti alle consegne del materiale in revisione previste dal committente; il *Riassunto delle attività di verifica* (A.1) contiene l'analisi del processo, corredate da delle considerazioni comprendenti le problematiche riscontrate;
- le modifiche al processo vengono attuate a partire dall'inizio del processo incrementale successivo. Tali attività sono programmate nel *Piano di Progetto v1.0.0*.



5 Pianificazione ed esecuzione del collaudo

Non avendo ancora affrontato la fase di progettazione del prodotto, allo stato attuale, non è possibile definire in dettaglio i collaudi. Pertanto la pianificazione ed esecuzione dei collaudi saranno trattate nella prossima revisione.

A Resoconto delle attività di verifica

A.1 Riassunto delle attività di verifica

A.1.1 Revisione dei requisiti

L'attività di verifica svolta dai *Verificatori* è avvenuta come pianificato nel *Piano di Progetto v1.0.0* al termine della stesura di ogni documento stilato. La verifica dei documenti è stata svolta seguendo le indicazioni delle *Norme di Progetto v1.0.0* e misurando le metriche indicate in 3.1. L'attività di *Walkthrough* ha evidenziato una serie di anomalie, in questo modo è stato possibile stilare la lista di anomalie frequenti che si potranno controllare tramite *Inspection* (fare riferimento alle *Norme di Progetto v1.0.0*).

A.2 Dettaglio delle verifiche tramite analisi

A.2.1 Analisi dei Requisiti di Massima

A.2.1.1 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante l'attività di *Analisi dei Requisiti di Massima* e il relativo esito basato sui range stabiliti in 3.1.1.

Documento	Valore indice	Esito
Analisi dei Requisiti v1.0.0	54	superato
Glossario v1.0.0	53	superato
Norme di Progetto v1.0.0	57	superato
Piano di Progetto v1.0.0	47	superato
Piano di Qualifica v1.0.0	50	superato
Studio di Fattibilità v1.0.0	45	superato

Tabella 2: Esiti del calcolo dell'indice Gulpease - *Analisi dei Requisiti di Massima*

A.2.1.2 Processi

Vengono qui riportati i valori degli indici *Schedule Variance* e *Budget Variance*, descritti nella sezione 3.2, per ogni documento redatto durante l'attività di *Analisi dei Requisiti di Massima*.

Documenti	SV	BV
Analisi dei Requisiti v1.0.0	0%	-4,79%
Glossario v1.0.0	0%	0%
Norme di Progetto v1.0.0	0%	10,99%
Piano di Progetto v1.0.0	0%	6,90%
Piano di Qualifica v1.0.0	0%	0,79%
Studio di Fattibilità v1.0.0	0%	0%

Tabella 3: Esiti del calcolo degli indici Schedule Variance e Budget Variance - *Analisi dei Requisiti di Massima*

B Qualità

B.1 Ciclo di Deming

La qualità perseguita nel presente documento si basa sugli standard ISO/IEC 15504 e ISO/IEC 9126 con l'obiettivo di approfondirne incrementalmente la copertura. Ogni processo segue il metodo PDCA che prevede l'iterazione ripetuta tra i quattro stadi, con l'obiettivo di incrementare la qualità dei processi di volta in volta. La sequenza logica dei quattro punti ripetuti garantisce un miglioramento continuo ed è la seguente:

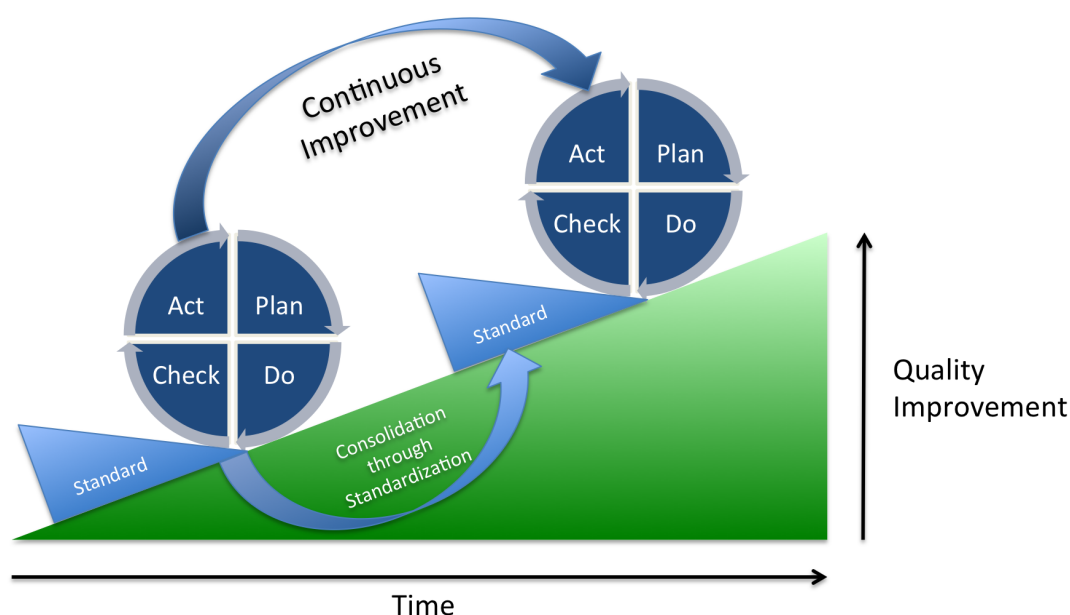


Figura 2: Continuous quality improvement with PDCA

- **Plan:** stabilire obiettivi e processi volti a raggiungere determinate finalità in accordo con i risultati attesi;
- **Do:** mettere in atto soluzioni e piani individuati nella fase precedente;
- **Check:** verificare che l'azione correttiva sia stata eseguita nei tempi previsti. Vanno considerate metriche come la *schedule variance* (vedi 3.2.1) e la completezza dei risultati attesi. Vanno confrontati i dati ottenuti con quelli di partenza tramite l'analisi di grafici e tabelle;
- **Act:** rendere prassi comune la soluzione trovata in modo da rendere consolidate e irreversibili le azioni correttive. Ultimata questa fase si procederà con una nuova iterazione a partire dalla fase di Plan.

B.2 Qualità dei processi

Lo $SPICE_G$, definito in ISO/IEC 15504, specifica che la qualità è collegata alla maturazione dei processi individuando dei livelli di maturità, ai quali il fornitore può riferirsi per definire le proprie capacità organizzative. Vengono definiti:

- **Modelli di riferimento:**
 - dimensione del processo;
 - livello di maturità dei processi:
 - * **5:** Ottimizzato
 - * **4:** Predicibile
 - * **3:** Stabilito
 - * **2:** Gestito
 - * **1:** Eseguito
 - * **0:** Incompleto

La capacità di un processo viene misurata tramite degli attributi che sono assimilabili alle metriche dei processi individuate in 3.2. In particolare la *Schedule Variance* permette di capire se un processo è incompleto o gestito; il gruppo giungerà a maturazione quando i processi diventeranno predicibili, ossia quando la *Schedule Variance* subirà al più lievi oscillazioni.

- **Stime** che si concretizzano in una struttura per la misurazione composta da:
 - i processi di misurazione, indicati nel *Piano di Progetto v1.0.0*;
 - un modello per la misurazione identificabile in questo documento;
 - gli strumenti utilizzati, specificati nelle *Norme di Progetto v1.0.0*.
- **Competenze e Qualifiche** di chi controlla; vengono definite dallo standard una serie di attività volte a formare chi opera l'attività di stesura del *Piano di Qualifica*. Tutti i componenti del gruppo si impegnano a studiare e ad applicare al meglio quanto descritto in questo documento.

B.3 Qualità del prodotto software

Specificata in ISO/IEC 9126 si suddivide in:

- **Quality model:** viene classificata la qualità del software secondo un set di caratteristiche che verranno approfondite nel corso del progetto:
 - *functionality*

- *reliability*
- *usability*
- *efficiency*
- *maintainability*
- *portability*

Allo stato attuale, il gruppo non è in grado di individuare l'insieme delle tecnologie utilizzate dal software.

- **External metrics:** sono le metriche rilevate tramite analisi dinamica, verranno specificate con il concretizzarsi della *Specifica Tecnica*;
- **Internal metrics:** sono le metriche rilevate in analisi statica specificate nella sezione 3;
- **Quality in use metrics:** sono metriche rilevabili allo stato di prodotto *usabile* in condizioni reali, si rimanda la definizione di tale aspetto a quando verranno trattate le considerazioni sull'usabilità del prodotto in uno scenario di utilizzo reale, questo deve avvenire non oltre la *Codifica*.