



Obelix Group
obelixswe@gmail.com

Piano di Qualifica

Versione	<i>v3_0_0</i>
Data creazione	2017-02-25
Redattori	Nicolò Rigato Silvio Meneguzzo
Verificatori	Emanuele Crespan
Approvazione	Riccardo Saggese
Stato	Approvato
Uso	esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Red Babel Gruppo Obelix

Sommario

Documento contenente le strategie adottate dal gruppo Obelix per raggiungere gli obiettivi qualitativi richiesti per il prodotto Monolith.



Diario delle revisioni

Versione	Modifica	Autore e Ruolo	Data
3.0.0	Approvazione documento	Riccardo Saggese Responsabile	2017-06-19
2.1.0	Verifica sezione "Resoconto delle attività di verifica"	Emanuele Crespan Verificatore	2017-06-19
2.0.1	Modifica sezione "Resoconto delle attività di verifica"	Nicolò Rigato Verificatore	2017-06-18
2.0.0	Approvazione documento	Nicolò Rigato Responsabile	2017-04-30
1.1.0	Verifica documento	Federica Schifano Verificatore	2017-04-29
1.0.3	Aggiunta sezione "Standard di Qualità" all'appendice	Silvio Meneguzzo Verificatore	2017-04-25
1.0.2	Aggiunto paragrafo "Inspection"	Silvio Meneguzzo Verificatore	2017-04-23
1.0.1	Aggiunto paragrafo "Walkthrough"	Silvio Meneguzzo Verificatore	2017-04-23
1.0.0	Approvazione documento	Nicolò Rigato Responsabile	2017-03-09
0.1.0	Verifica del documento	Silvio Meneguzzo Verificatore	2017-03-09
0.0.6	Stesura sezione Resoconto delle attività di verifica	Tomas Mali Verificatore	2017-03-05
0.0.5	Stesura sezione Gestione amministrativa della revisione	Riccardo Saggese Verificatore	2017-03-04
0.0.4	Stesura sezione Strategie di Verifica	Tomas Mali Verificatore	2017-02-28
0.0.3	Stesura sezione Definizione obiettivi di qualità	Riccardo Saggese Verificatore	2017-02-27
0.0.2	Stesura sezione Introduzione	Tomas Mali Verificatore	2017-02-26
0.0.1	Creazione template	Nicolò Rigato Responsabile	2017-02-25

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Normativi	4
1.4.2	Informativi	4
2	Strategia di verifica	5
2.1	Definizione obiettivi	5
2.1.1	Qualità di processo	5
2.1.2	Qualità di prodotto	5
2.2	Procedure di controllo della qualità di processo	6
2.3	Procedure di controllo della qualità di prodotto	6
2.4	Organizzazione	6
2.4.1	Analisi	6
2.4.2	Progettazione Architettuale	7
2.4.3	Progettazione di Dettaglio e Codifica	7
2.5	Responsabilità	7
2.6	Risorse	7
2.7	Tecniche di analisi	7
2.7.1	Analisi statica	7
2.7.2	Analisi dinamica	8
2.8	Misure e metriche	8
2.8.1	Metriche per i processi	8
2.8.2	Metriche per i prodotti	10
3	Gestione amministrativa della revisione	14
3.1	Comunicazione delle anomalie	14
3.2	Procedure di controllo per la qualità di processo	15
A	Standard di qualità	16
A.1	Standard ISO/IEC 15504	16
A.2	Ciclo di Deming PDCA	17
A.3	Standard ISO/IEC 9126	18
B	Resoconto delle attività di verifica	20
B.1	Revisione dei Requisiti	20
B.1.1	Processi	20
B.1.2	Documenti	20
B.2	Revisione di Progettazione	20
B.2.1	Processi	21
B.2.2	Documenti	21



Elenco delle tabelle

2	Tabella risultati test Gulpease	20
3	Tabella risultati Budget Variance / Schedule Variance	21
4	Tabella risultati test Gulpease	21



1 Introduzione

1.1 Scopo del documento

Lo scopo principale di questo documento è di ottenere una buona qualità, intesa non solo come qualità di prodotto, ma anche come qualità di processo. Per raggiungere tale obiettivo è necessario un lavoro intellettuale e arduo poiché la qualità del software è diverso dagli altri prodotti manifatturieri ed è difficile misurarla in maniera oggettiva. Per rilevare e successivamente correggere le anomalie in modo efficace è necessaria un'attività costante di verifica e validazione da parte del *team_{|G|}* Obelix.

1.2 Scopo del prodotto

Lo scopo del prodotto è quello di creare un *SDK_{|G|}* che permetta la realizzazione di cosiddette "bolle interattive" di diversi tipi in base alle richieste di utenti (che nel nostro caso saranno gli sviluppatori). Il prodotto si completerà con la realizzazione di una demo presentabile mediante una web app con obiettivo di testare e provare il corretto funzionamento del SDK.

1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario xxx*

1.4 Riferimenti

1.4.1 Normativi

- **Norme di progetto:** *Norme di Progetto xxx*:
- **Capitolato d'appalto C5:** *RedBabel, Monolith* <http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C5.pdf/>

1.4.2 Informativi

- **Piano di Progetto:** *Piano di Progetto*
- **PDCA (Plan-Do-Check-Act):** <http://it.wikipedia.org/wiki/PCDA>
- **Standard ISO/IEC 12207:2008-IEEE Std 12207-2008:** <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4475822>
- **Standard ISO/IEC 15504:** http://en.wikipedia.org/wiki/ISO/IEC_15504/
- **Standard ISO/IEC 9126:** http://it.wikipedia.org/wiki/ISO/IEC_9126
- **Indice di Gulpease:** http://it.wikipedia.org/wiki/Indice_Gulpease



2 Strategia di verifica

Con le strategie utilizzate si cerca di automatizzare il lavoro di verifica. Lo scopo è quello di ottenere un riscontro affidabile ed adeguato per assicurare un grado di qualità predeterminato e ridurre il lavoro manuale permettendo così una validazione semplificata.

2.1 Definizione obiettivi

2.1.1 Qualità di processo

La qualità di processo è definita dallo standard ISO/IEC 15504 (SPICE). Questo standard specifica come la qualità è collegata alla maturazione dei processi. Non sarebbe possibile definire la qualità del prodotto senza garantire la qualità del processo. La qualità del prodotto nasce dunque dalla qualità del processo. Per garantire tutto ciò il team Obelix fa uso dello standard ISO/IEC 15504 denominato SPICE, il quale fornisce gli strumenti necessari a valutare l'idoneità dei processi.

Miglioramento costante Per quantificare la capacità dei processi e porsi obiettivi quantitativi di miglioramento abbiamo deciso di adottare il metodo PDCA. Per una descrizione più dettagliata del metodo PDCA si faccia riferimento all'appendice *B.1*

Rispetto della pianificazione Per capire se le attività di un processo sono in ritardo rispetto a quanto pianificato all'interno del documento *Piano di Progetto vxxx* viene utilizzata la metrica Schedule Variance. Si desidera che il ritardo accumulato sia minore del 5% rispetto al totale pianificato. Sarebbe invece ottimale essere esattamente in linea con quanto prevede il *Piano di progetto vxxx*, o essere addirittura in anticipo.

Rispetto del budget Viene utilizzata la metrica Budget Variance Per capire se i costi di un processog rientrano nel budget previsto dal *Piano di progetto vxxx*. L'obiettivo minimo è quello di avere dei costi che non superano il budget a disposizione per più del 10%. Sarebbe invece ottimale che i costi fossero esattamente in linea con il preventivo o addirittura minori.

2.1.2 Qualità di prodotto

Per garantire la qualità del prodotto il team Obelix cercherà di aderire al meglio allo standard di qualità ISO/IEC 9126.

Qualità dei documenti Gli obiettivi della qualità che riguardano i documenti ai quali il team Obelix desidera raggiungere sono i seguenti:

- i documenti devono essere comprensibili da individui dotati di una licenza superiore
- i documenti devono essere corretti a livello ortografico
- i documenti non devono contenere concetti errati



Leggibilità e comprensibilità Per la leggibilità e comprensibilità dei documenti da persone dotate di una licenza superiore viene utilizzato l'indice Gulpease. È desiderabile che i documenti abbiano costantemente un indice maggiore a 40 come soglia accettabile e 60 come ottimale. Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica "Indice di Gulpease" alla sezione 2.8.2.

Correttezza ortografica Per capire quanto i documenti siano effettivamente corretti a livello ortografico viene utilizzata la metrica "Percentuale di errori ortografici rinvenuti e non corretti". Si desidera che tutti gli errori ortografici che sono stati trovati siano corretti. Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica "Errori ortografici rinvenuti e non corretti" alla sezione 2.8.2.

Correttezza concettuale Per capire quanto i documenti siano effettivamente corretti a livello concettuale viene utilizzata la metrica "Percentuale di errori concettuali rinvenuti e non corretti". L'obiettivo ottimale è quello di correggere tutti gli errori trovati. Per una descrizione dettagliata della metrica utilizzata si faccia riferimento alla metrica "Errori concettuali rinvenuti e non corretti" alla sezione 2.8.2.

2.2 Procedure di controllo della qualità di processo

Le linee guida per la gestione della qualità di processo seguono il modello $PDCA|G|$ descrivendo 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 delle qualità dei processi

2.3 Procedure di controllo della qualità di prodotto

Con i seguenti processi verrà garantita il controllo della qualità del prodotto:

- **Software Quality Assurance (SQA)**: è l'insieme delle attività che serve per garantire il raggiungimento degli obiettivi di qualità
- **Verifica**: assicura che non siano stati introdotti errori nel prodotto con l'esecuzione dei processi
- **Validazione**: è la conferma oggettiva che assicura che i prodotti finali soddisfino i requisiti e le aspettative attese

2.4 Organizzazione

2.4.1 Analisi

In questo periodo verrà verificata la corrispondenza tra i casi d'uso e i requisiti. Verrà inoltre controllato il rispetto dei processi e della documentazione prodotta.



2.4.2 Progettazione Architettuale

In questo periodo avviene la verifica dei processi relativi all'analisi e ai nuovi documenti di progettazione. Inoltre si verifica che i test siano adeguatamente pianificati come descritti ed eseguiti nel documento *Norme di Progetto vxxx*

2.4.3 Progettazione di Dettaglio e Codifica

In questo periodo avviene la verifica dei processi relativi alla progettazione insieme alla verifica delle attività di codifica tramite tecniche di analisi statica e dinamica.

2.5 Responsabilità

La responsabilità delle verifiche è attribuita al *Responsabile di progetto* e ai *Verificatori*. All'interno del *Piano di Progetto vxxx* sono definiti i compiti e le modalità di attuazione.

2.6 Risorse

La raggiunta degli obiettivi qualitativi richiede sia risorse umane che tecnologiche. Hanno maggiore responsabilità per l'attività di verifica e validazione il *Responsabile del progetto* e il *Verificatore*. Per una dettagliata descrizione dei ruoli e delle loro responsabilità bisogna fare riferimento alle *Norme di Progetto vxxx*. Per risorse tecniche e tecnologiche si intendono tutti gli strumenti software e hardware che il gruppo Obelix intende utilizzare per attuare le attività di verifica su processi e prodotti. Per una dettagliata e accurata descrizione di tali strumenti si faccia riferimento nelle *Norme di Progetto vxxx*.

2.7 Tecniche di analisi

2.7.1 Analisi statica

L'analisi statica è una tecnica di analisi che si applica sia alla documentazione che al codice e permette di individuare errori ed anomalie. Essa si svolge in due modi distinti che sono Walkthrough ed Inspection.

Walkthrough È una tecnica che viene utilizzata soprattutto nelle prime attività del progetto quando ancora non è presente una adeguata esperienza da parte dei membri del gruppo, che permetta di attuare una verifica più mirata e precisa. Con l'utilizzo di questa tecnica, il *Verificatore* sarà in grado di stilare una lista di controllo con gli errori più frequenti in modo da favorire il miglioramento di tale attività nelle fasi future. Walkthrough è un'attività onerosa e collaborativa che richiede l'intervento di più persone per essere efficace ed efficiente. Dopo una prima fase di lettura e individuazione degli errori, segue una fase di discussione con la finalità di esaminare i difetti riscontrati e di proporre le dovute correzioni. L'ultima fase consiste nel correggere gli errori rilevati e nello scrivere un rapporto che elenchi le modifiche effettuate.



Inspection L'inspection consiste nell'analisi mirata di alcune parti del documenti o del codice ritenute fonti maggiori di errore. Deve essere seguita una lista di controllo per svolgere efficacemente questa attività; tale lista deve essere redatta anticipatamente ed è sostanzialmente frutto dell'esperienza maturata dai membri del team con tecniche di walkthrough. L'inspection è dunque più rapida del walkthrough, in quanto il documento viene analizzato solo in alcune sue parti e con una lista di controllo ben precisa.

2.7.2 Analisi dinamica

L'analisi dinamica si applica solamente al prodotto software e viene svolta durante l'esecuzione del codice mediante l'uso di test progettati per rilevare la presenza di difetti nel software. L'obiettivo del test del software è infatti, quello di realizzare un prodotto il più possibile esente da errori. Il principale ostacolo alla fase di test è sintetizzato nella tesi di Dijkstra, la quale afferma che il test può indicare la presenza di errori, ma non ne può garantire l'assenza. Affinchè tale attività sia utile e generi risultati attendibili è necessario che i test effettuati siano ripetibili: dato un certo input deve essere prodotto sempre uno stesso output in uno specifico ambiente. Di conseguenza, i tre elementi fondamentali di un test sono:

- **Ambiente:** sistema hardware e software sui quali è stato pianificato l'utilizzo del prodotto software sviluppato. Su di essi deve essere specificato uno stato iniziale dal quale poter eseguire il test
- **Specifici:** definizione di input e output
- **Procedure:** definizione di come devono essere svolti i test, in che ordine devono essere eseguiti e come devono essere analizzati i risultati

2.8 Misure e metriche

Con lo scopo di poter monitorare in modo consapevole l'andamento dei processi e la qualità del prodotto si utilizzano delle metriche per rendere misurabili e valutabili in modo oggettivo alcune caratteristiche di documenti, processi e software.

2.8.1 Metriche per i processi

Le metriche per i processi hanno lo scopo di monitorare e rendere prevedibile l'andamento delle variabili di maggior criticità del progetto: tempo e costo. Le metriche sono utilizzate in modo consultivo per consentire 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*.

Schedule Variance Permette di calcolare le tempistiche rispetto l'organizzazione delle attività pianificate alla data corrente. È un indicatore di efficacia soprattutto a beneficio del 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

Budget Variance Permette di calcolare i costi rispetto alla data corrente. È un indicatore che ha un valore unicamente contabile e 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 speso è minore di quanto pianificato
- $BV < 0$: il budget speso è maggiore di quanto pianificato
- $BV = 0$: il budget speso è in linea con quanto stabilito

Produttività

Produttività di documentazione Indica la produttività media nel redigere i documenti.

$$\text{Produttività di documentazione} = \frac{\text{Parole}}{\text{Ore persona}}$$

Dove:

- **Parole:** indica il numero di parole presente nei documenti
- **Ore persona:** indica il numero di ore produttive impiegate per realizzare tali documenti

Parametri utilizzati

- Range-ottimale: $[\geq 100]$



Produttività di test Indica la produttività media nell'effettuare test.

$$\text{Produttività di test} = \frac{\text{Numero di test}}{\text{Ore persona}}$$

Dove:

- **Numero di test:** indica il numero di test eseguiti
- **Ore persona:** indica il numero di ore produttive impiegate per l'esecuzione di tali test

Produttività di codifica Indica la produttività media nelle attività di codifica.

$$\text{Produttività di codifica} = \frac{\text{LOCs}}{\text{Ore persona}}$$

Dove:

- **LOCs:** indica il numero di linee di codice prodotte (Lines Of Code)
- **Ore persona:** indica il numero di ore produttive impiegate per produrre tale codice

Copertura del test Indica la percentuale di casi coperti dai test eseguiti.

$$\text{Copertura del test} = \frac{\text{Numero di funzioni testate} \times 100}{\text{Numero totale di funzioni disponibili}}$$

Parametri utilizzati

- Range-accettazione: $[70 - 100]$
- Range-ottimale: $[80 - 100]$

2.8.2 Metriche per i prodotti

Metriche per i documenti I documenti sono di qualità se sono leggibili. La leggibilità è misurabile tramite un indice calcolabile sulle caratteristiche del testo.

Indice Gulpease L'indice Gulpease misura la leggibilità di un testo in Italiano utilizzando variabili linguistiche come la lunghezza delle parole e delle frasi.

$$G = 89 + \frac{300 \times \text{numero delle frasi} - 10 \times \text{numero delle lettere}}{\text{numero delle parole}}$$

Il valore risultante è un numero compreso tra 0 e 100, dove 100 indica la leggibilità massima e 0 la leggibilità minima. I documenti sono da considerare secondo le seguenti fasce:

- $G < 80$ sono considerati difficili per chi ha la sola licenza elementare
- $G < 60$ sono considerati difficili per chi ha la sola licenza media



- $G < 40$ sono considerati difficili per chi ha un diploma di scuola superiore

Parametri utilizzati

- Range di accettazione [40-90]
- Range ottimale [50-90]

Errori ortografici rinvenuti e non corretti Questa metrica è necessaria per capire quanto un documento sia corretto dal punto di vista ortografico. Gli strumenti automatici possono trovare la maggior parte degli errori ortografici all'interno di un testo ma non sono molto attendibili poiché si basano sul numero degli errori rinvenuti ma non successivamente corretti. Vengono di seguito riportati i range stabiliti per quest'ultima metrica:

- una percentuale di errori non corretti maggiore allo 0% è ritenuta negativa
- una percentuale di errori non corretti pari allo 0% è ritenuta accettabile
- una percentuale di errori non corretti minore allo 0% è ritenuta ottimale

Errori concettuali rinvenuti e non corretti Questa metrica è necessaria per capire quanto un documento sia corretto dal punto di vista concettuale. Vengono di seguito i range stabiliti per la metrica appena introdotta:

- una percentuale di errori non corretti maggiore al 5% è ritenuta negativa
- una percentuale di errori non corretti minore del 5% è ritenuta accettabile
- una percentuale di errori non corretti pari allo 0% è ritenuta ottimale

Metriche per software

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. Nel grafo sopracitato i nodi corrispondono a gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo di istruzioni può essere eseguito immediatamente dopo il primo gruppo. Tale indice può essere applicato indistintamente a singole funzioni, moduli, metodi o *package*_{|G|} di un programma. Si vuole utilizzare tale metrica per limitare la complessità durante le attività di sviluppo del prodotto software. Può rivelarsi utile durante il testing per determinare il numero di test necessari, essendo l'indice di complessità un limite superiore al numero di test necessari per raggiungere la copertura completa.

Parametri utilizzati:

- Range-accettazione: [0 – 25]
- Range-ottimale: [0 – 10]



Numero di metodi - NOM Il *Number of methods* è una metrica usata per calcolare la media delle occorrenze dei metodi per package. Un package non dovrebbe contenere un numero eccessivo di metodi. Valori superiori al range ottimale massimo potrebbero indicare una necessità di maggiore scomposizione del package.

Parametri utilizzati:

- Range-accettazione: $[3 - 10]$
- Range-ottimale: $[3 - 7]$

Variabili non utilizzate e non definite La presenza di variabili non utilizzate viene considerata pollution, pertanto non viene tollerata. Tali occorrenze vengono rilevate analizzando l'Abstract syntax tree eseguendo un confronto tra le variabili dichiarate e quelle inizializzate. Per sua natura, *Javascript*_{|G|} non blocca l'insorgenza di tali occorrenze, pertanto si rischia di dichiarare una variabile e poi utilizzarne una con nome leggermente diverso, oppure semplicemente dichiarare una variabile che in seguito non verrà mai utilizzata.

Parametri utilizzati:

- Range-accettazione: $[0 - 0]$
- Range-ottimale: $[0 - 0]$

Numero parametri per metodo Un numero elevato di parametri per un metodo potrebbe evidenziare un metodo troppo complesso. Non c'è una regola forte per il numero di parametri possibili in un metodo, ma citando Robert Martin in Clean Code¹ :

The ideal number of arguments for a function is zero (niladic). Next comes one (monadic), followed closely by two (dyadic). Three arguments (triadic) should be avoided where possible. More than three (polyadic) requires very special justification – and then shouldn't be used anyway.

Vengono quindi seguite le linee guida dei seguenti parametri:

- Range-accettazione: $[0 - 8]$
- Range-ottimale: $[0 - 4]$

Halstead La metrica di *Halstead*_{|G|} oltre ad essere un indice di complessità, permette di identificare le proprietà misurabili del software e le relative relazioni. Si basa sulla necessità che una metrica sia indipendente dalla specifica piattaforma e dal linguaggio di programmazione. Sono identificati i seguenti dati all'interno di un problema astratto:

- n_1 : indica il numero di operatori distinti

¹Robert Martin, Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall (2008)



- n_2 : indica il numero di operandi distinti
- N_1 : indica il numero totale di operatori
- N_2 : indica il numero totale di operandi

Da cui si ottiene:

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

Data la scarsa disponibilità in rete di valori di riferimento per il Javascript, i range saranno da valutare in un momento successivo alla RR.

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

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

Parametri utilizzati:

- Range-accettazione: $[0 - 30]$
- Range-ottimale: $[0 - 15]$

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.

$$V = N \times \log_2 n$$

Parametri utilizzati:

- Range-accettazione: $[20 - 1500]$
- Range-ottimale: $[20 - 1000]$

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.

$$E = V \times D$$

Parametri utilizzati:

- Range-accettazione: $[0 - 400]$
- Range-ottimale: $[0 - 300]$



Maintainability index Questa metrica² è una scala logaritmica da $-\infty$ a 171, calcolata sulla base delle linee di codice logiche, della complessità ciclomatica e dall'indice Halstead effort. Un valore alto indica una maggiore manutenibilità.

Parametri utilizzati:

- Range-accettazione: $[> 70]$
- Range-ottimale: $[> 90]$

Statement Coverage Permette di calcolare quante linee di codice di ciascun modulo delle unità sono eseguite almeno una volta nell'esecuzione dei test. Tale metrica è espressa in percentuale.

Parametri utilizzati:

- Range-accettazione: $[70 - 100]$
- Range-ottimale: $[85 - 100]$

Branch Coverage Permette di calcolare quanti rami della logica di flusso sono attraversati almeno una volta durante l'esecuzione dei test. Tale metrica è espressa in percentuale.

Parametri utilizzati:

- Range-accettazione: $[70 - 100]$
- Range-ottimale: $[85 - 100]$

3 Gestione amministrativa della revisione

3.1 Comunicazione delle anomalie

Identificare le anomalie permette la correzione dei difetti ricercati dal processo di Software Quality Management e informa il *Responsabile* di progetto sullo stato del prodotto. Analizzare e catalogare le anomalie è utile per discutere, durante revisioni e riunioni, su quali modifiche e correzioni applicare e con quale priorità. Di seguito è presente la lista delle definizioni di anomalie (IEEE 610.12-90) adottate dal gruppo:

- **Error:** differenza riscontrata tra il risultato di una computazione e il valore teorico atteso
- **Fault:** un passo, un processo o un dato definito in modo erroneo. Corrisponde a quanto viene comunemente definito come bug
- **Failure:** il risultato di un fault
- **Mistake:** azione umana che produce un risultato errato

²Definita nel 1991 da Paul Oman e Jack Hagemeister alla University of Idaho.



3.2 Procedure di controllo per la qualità di processo

Le procedure di controllo per la qualità di processo hanno il fine di migliorare la qualità del prodotto e diminuire i costi e tempi di sviluppo. Esistono due approcci principali:

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

Il team adotterà il primo approccio, essendo più adatto ad un gruppo inesperto. Con una visione proattiva si cerca di avere maggior controllo e previsione sulle attività da svolgere. Questa viene anche indicata come best practice per gruppi poco esperti. Il processo con maggiore influenza sulla qualità del sistema non è quello di sviluppo ma quello di progettazione. È qui che 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 del processo:** misura gli attributi del progetto, punta ad allineare gli obiettivi con le misurazioni effettuate. Questo forma una *baseline*_[G] che aiuta a capire se i miglioramenti hanno avuto effetto
- **Analisi del processo:** vengono identificate le problematiche ed i colli di bottiglia dei processi
- **Modifiche del processo:** i cambiamenti vengono proposti in risposta alle problematiche riscontrate

Il team procederà nel seguente modo:

- Nell'appendice del seguente documento verranno inserite le misurazioni rilevate sulle metriche descritte nella sezione 2.8.1
- Le modifiche al processo vengono attuate all'inizio del processo incrementale successivo. Queste attività sono programmate nel *Piano di Progetto xxx*



A Standard di qualità

Vengono riportati gli obbiettivi di qualità che il team Obelix dovrebbe raggiungere durante lo svolgimento del progetto. Il team adotta degli standard, modelli e metriche per convincersi se un obiettivo è stato raggiunto o meno.

A.1 Standard ISO/IEC 15504

La qualità di processo è definita dallo standard ISO/IEC 15504 (SPICE). Questo standard specifica come la qualità è collegata alla maturazione dei processi. Non sarebbe possibile definire la qualità del prodotto senza garantire la qualità del processo. La qualità del prodotto nasce dunque dalla qualità del processo. Per garantire tutto ciò il team Obelix fa uso dello standard ISO/IEC 15504 denominato SPICE, il quale fornisce gli strumenti necessari a valutare l'idoneità dei processi. Per l'esattezza sono previsti sei livelli di maturità:

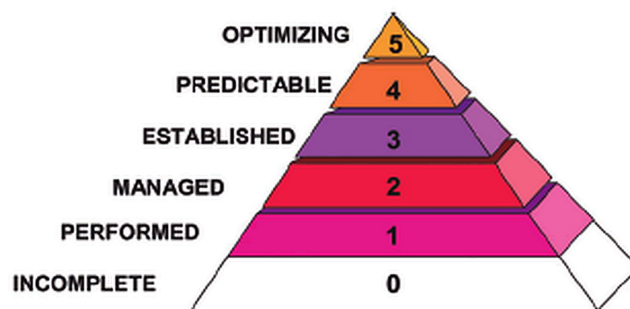


Figura 1: Standar ISO/IEC 15504

- **Livello 0 - Incomplete process**
Il processo non è implementato o non riesce a raggiungere i suoi obiettivi
- **Livello 1 - Performed process**
Il processo viene messo in atto e raggiunge i suoi scopi
- **Livello 2 - Managed process**
Il processo viene eseguito sulla base di obiettivi ben definiti
- **Livello 3 - Established process**
Il processo viene eseguito in base ai principi dell'ingegneria del software
- **Livello 4 - Predictable process**
Il processo è attuato all'interno di limiti ben definiti
- **Livello 5 - Optimizing process**
Il processo è predicibile ed è in grado di adattarsi per raggiungere obiettivi specifici e rilevanti

Per valutare il livello di maturità di un processo si utilizzano le metriche descritte nella sezione 2.8.1, in particolare lo *schedule variance* aiuta a capire quando un processo raggiunge uno stato accettabile se il valore subisce al massimo lievi oscillazioni da quanto previsto.



A.2 Ciclo di Deming PDCA

Il ciclo di Deming (o PDCA) è un approccio per il miglioramento continuo come parte integrante della gestione della qualità. Il ciclo di Deming è suddiviso in quattro fasi:

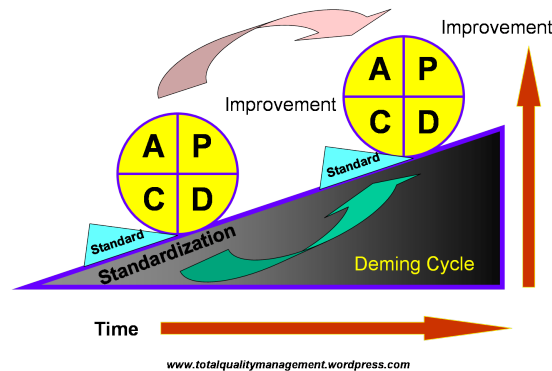


Figura 2: Ciclo di Deming

- **Plan** - Consiste nella definizione di ciò che deve essere fatto per risolvere un problema o migliorare un processo e nella successiva pianificazione delle azioni da svolgere
 - Precisazione degli obiettivi del miglioramento da attuare
 - Raccolta dei dati relativi al problema o processo
 - Mappatura del processo utilizzando un diagramma di flusso od altro strumento utile
 - Individuazione delle cause principali dei vincoli da rimuovere
 - Determinazione degli interventi necessari per risolvere la situazione
 - Determinazione dei risultati attesi
 - Definizione delle responsabilità per la fase di attuazione
 - Pianificazione delle azioni da svolgere
 - Pianificazione delle risorse
 - Determinazione delle metriche per misurare i miglioramenti o gli scostamenti da quanto previsto
- **Do** - Durante questa fase del PDCA, vengono attuate le soluzioni ed i piani precedentemente definiti
 - I responsabili individuati mettono in pratica le azioni previste
 - Ogni soluzione è implementata per un periodo di prova
 - Viene verificata l'adeguatezza delle soluzioni adottate rispetto agli obiettivi attesi
 - Vengono formati i dipendenti sulle nuove modalità operative a fronte delle soluzioni adottate



- **Check** - Successivamente alla prima applicazione delle soluzioni progettate, queste vengono sottoposte ad un controllo nel tempo per verificare la sostenibilità di quanto realizzato ed eventualmente approfondire cosa non ha funzionato.

In questa fase del PDCA occorre riprendere le attività di analisi delle cause dei problemi per individuare cosa deve essere fatto per superare le difficoltà o le anomalie che ancora persistono

- **Act** - Una volta che le soluzioni adottate hanno dimostrato di funzionare, è opportuno procedere a
 - Standardizzare il miglioramento ottenuto applicandolo in via definitiva
 - Individuare eventuali esigenze di formazione del personale per rendere operative le soluzioni adottate
 - Continuare a monitorare la situazione ripetendo il ciclo più volte fino a raggiungere i miglioramenti desiderati
 - Individuare altre opportunità di miglioramento

A.3 Standard ISO/IEC 9126

Aderendo a questo standard il team si impegna a garantire nel prodotto le seguenti qualità (definite con relativa metrica, misura e strumenti di controllo):

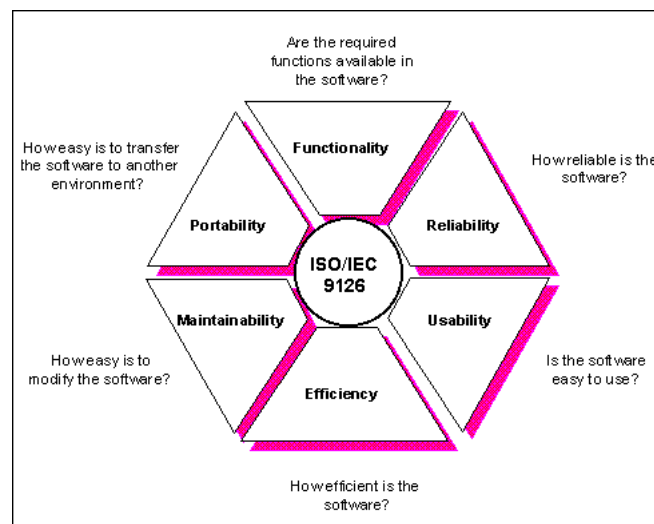


Figura 3: Standar ISO/IEC 9126

1. **Funzionalità**: il sistema prodotto deve garantire tutte le funzionalità indicate nel documento *Analisi dei Requisiti vxrx*. L'implementazione dei requisiti deve essere più completa possibile
 - **Misura**: l'unità di misura usata sarà la quantità di requisiti mappati sulle componenti del sistema create e funzionanti



- **Metrica:** la sufficienza è stabilita nel soddisfacimento di tutti i requisiti obbligatori
 - **Strumenti:** per soddisfare questa qualità il sistema deve superare tutti i test previsti. Per informazioni dettagliate sugli strumenti si veda *Norme di Progetto vxxx*
2. **Affidabilità:** il sistema deve dimostrarsi il più possibile robusto e di facile ripristino in caso di errori
- **Misura:** l'unità di misura utilizzata sarà la quantità di esecuzioni del sistema andate a buon fine
 - **Metrica:** le esecuzioni dovranno spaziare il più possibile sulle varie parti del codice
 - **Strumenti:** si vedano le metriche adottate per la copertura a
3. **Usabilità:** il sistema prodotto deve risultare di facile utilizzo per la classe di utenti a cui è destinato. Tale sistema deve essere facilmente apprendibile e allo stesso tempo deve soddisfare tutte le necessità dell'utente.
- **Misura:** l'unità di misura usata sarà una valutazione soggettiva dell'usabilità. Questo è dovuto all'inesistenza di una metrica oggettiva adatta allo scopo
 - **Metrica:** non esiste una metrica adeguata che determinerà la sufficienza su questa qualità
 - **Strumenti:** si vedano le *Norme di Progetto vxxx*
4. **Efficienza:** il sistema deve fornire tutte le funzionalità nel più breve tempo possibile, riducendo al minimo l'utilizzo di risorse
- **Misura:** il tempo che lo sviluppatore impiegherà per la creazione di una bolla interattiva
 - **Metrica:** non è possibile definire una soglia oggettiva di sufficienza perché non è possibile valutare ogni possibile casistica di utilizzo
 - **Strumenti:** si vedano le *Norme di Progetto vxxx*
5. **Manutenibilità:** il sistema deve essere comprensibile ed estensibile in modo facile e verificabile
- **Misura:** le metriche sul codice descritte nella sezione 2.8.2 costituiscono l'unità di misura
 - **Metrica:** il software avrà le caratteristiche di manutenibilità descritte. Questo sarà possibile se il prodotto avrà la sufficienza in tutte le metriche
 - **Strumenti:** si vedano le *Norme di Progetto vxxx*
6. **Portabilità:** il sistema deve essere il più portabile possibile, in particolare dentro a Rocket.chat
- **Misura:** il front end deve rispettare gli standard W3C



- **Metrica:** il software avrà le caratteristiche di portabilità descritte. Questo sarà possibile se il prodotto avrà la sufficienza in tutte le metriche, descritte nella sezione 2.8.2
- **Strumenti:** si vedano le *Norme del Progetto vxxx*

B Resoconto delle attività di verifica

B.1 Revisione dei Requisiti

L'attività di verifica svolta dai *Verificatori* è avvenuta come determinato dal *Piano di Progetto vxxx* al termine della stesura di ogni documento previsto. La verifica svolta sui documenti e sui processi è avvenuta seguendo le indicazioni delle *Norme di Progetto vxxx* e misurando le metriche indicate nella sezione 2.8.2

B.1.1 Processi

In questo periodo è stata svolta un'attività di walkthrough non avendo gli elementi per effettuare l'attività di inspection. Nella verifica dei documenti sono stati riscontrati soprattutto errori grammaticali e di battitura dovuti a disattenzioni durante la stesura. Non è stato possibile eseguire nessun ciclo PDCA in mancanza di misurazioni precedenti sui processi.

B.1.2 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante l'analisi e relativo esito basato sui range stabiliti nella sezione 2.8.2

Documento	Valore indice	Esito
<i>Analisi dei Requisiti v1.0.0</i>	70	superato
<i>Norme di Progetto v1.1.0</i>	58	superato
<i>Piano di Progetto v1.1.0</i>	65	superato
<i>Piano di Qualifica v1.0.0</i>	65	superato
<i>Studio di Fattibilità v1.0.0</i>	67	superato

Tabella 2: Tabella risultati test Gulpease

B.2 Revisione di Progettazione

In questo periodo (antecedente la consegna di tale revisione) sono stati verificati i documenti ed i processi.

- Sono stati verificati i documenti applicando la procedura descritta nel documento *Norme di progetto vxxx*
- È stata applicata *l'analisi statica* secondo i criteri e le modalità indicata alla sezione 2.7.1.
- Si è applicato il ciclo PDCA per rendere più efficiente ed efficace il processo di verifica.
- Si sono calcolate per i documenti le metriche descritte nella sezione 2.8.



- Il tracciamento (requisiti - componenti) è stato effettuato tramite un database MySQL creato ad uso interno.
- L'avanzamento dei processi è stato controllato e verificato secondo le metodiche descritte nelle *Norme di Progetto xxx*.

B.2.1 Processi

Vengono qui riportati gli esiti delle verifiche sui processi produttivi.

Budget Variance / Schedule Variance

Attività	SV	BV
<i>Analisi dei Requisiti 2.0.0</i>	0€	0€
<i>Norme di Progetto 3.0.0</i>	+10€	+70€
<i>Piano di Progetto 3.0.0</i>	0€	0€
<i>Definizione di prodotto 1.0.0</i>	+25€	-110€
<i>Piano di qualifica 3.0.0</i>	-10€	0€
<i>Glossario 2.0.0</i>	0€	0€

Tabella 3: Tabella risultati Budget Variance / Schedule Variance

Complessivamente, la fase di Revisione di Progettazione ha:

- $SV = 25€$
- $BV = -40€$

Da valori di tali indici possiamo dedurre che:

- La data di fine della Revisione di Progettazione si è dimostrata essere 1 giorno prima di quella pianificata nel Piano di Progetto v3.0.0 . Aumentando il numero di ore di lavoro giornaliero ed utilizzando tutti gli slack inseriti durante la pianificazione il gruppo ha compresso i tempi con conseguente SV positivo;
- Il limite inferiore di accettabilità del BV è pari a -436€. Il valore ottenuto è quindi accettabile. Come già precisato, il gruppo per portare a termine gli obiettivi entro le date imposte ha aumentato il numero di ore di lavoro giornaliero. Questo ha quindi causato un aumento del BV in quanto le ore complessive di lavoro sono state vicine a quelle preventivate, ma il periodo di lavoro è stato compresso.

B.2.2 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante l'analisi e relativo esito basato sui range stabiliti nella sezione 2.8.2

Documento	Valore indice	Esito
<i>Analisi dei Requisiti 2.0.0</i>	60.1	superato
<i>Norme di Progetto 3.0.0</i>	56.4	superato
<i>Piano di Progetto 3.0.0</i>	67.5	superato
<i>Piano di Qualifica 3.0.0</i>	61.4	superato
<i>Definizione di Prodotto 1.0.0</i>	76.9	superato



Tabella 4: Tabella risultati test Gulpease