



Piano di Qualifica

7DOS - 1 Gennaio 2019

Informazioni sul documento

Versione	1.0.0
Responsabile	Marco Costantino
Verifica	Lorenzo Busin Michele Roverato
Redazione	Giacomo Barzon Giovanni Sorice Nicolò Tartaggia Andrea Trevisin
Stato	Approvato
Uso	Esterno
Destinato a	Prof.Tullio Vardanega Prof.Riccardo Cardin Zucchetti 7DOS
Email	7dos.swe@gmail.com

Descrizione

Questo documento descrive le operazioni di verifica e validazione relative al progetto *G&B*.

Diario delle modifiche

Modifica	Autore	Ruolo	Data	Versione
<i>Approvazione del documento</i>	Marco Costantino	Responsabile	2019-01-01	1.0.0
<i>Verifica del documento</i>	Michele Roverato	Verificatore	2019-01-01	0.5.0
<i>Verifica del documento</i>	Lorenzo Busin	Verificatore	2018-12-29	0.4.0
<i>Completamento stesura Resoconto</i>	Giovanni Sorice	Analista	2018-12-28	0.3.1
<i>Inizio stesura Resoconto</i>	Giovanni Sorice	Analista	2018-12-26	0.3.0
<i>Completamento stesura Strategia generale</i>	Nicolò Tartaggia	Analista	2018-12-13	0.2.2
<i>Completamento stesura Metriche</i>	Andrea Trevisin	Analista	2018-12-13	0.2.1
<i>Inizio stesura Metriche</i>	Giacomo Barzon	Analista	2018-12-05	0.2.0
<i>Inizio stesura Strategia generale</i>	Nicolò Tartaggia	Analista	2018-12-05	0.1.0
<i>Stesura della sezione Introduzione</i>	Andrea Trevisin	Analista	2018-12-05	0.0.2
<i>Stesura dello scheletro del documento</i>	Giacomo Barzon	Analista	2018-12-05	0.0.1

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Maturità del documento	4
1.5	Riferimenti	5
1.5.1	Normativi	5
1.5.2	Informativi	5
2	Strategia generale per la verifica	6
2.1	Obiettivi	6
2.2	Qualità di processo	6
2.2.1	ISO/IEC 15504(SPICE)	6
2.2.2	Ciclo di miglioramento continuo (PDCA)	7
2.2.3	Gestione	8
2.3	Qualità di prodotto	8
2.3.1	Functional Suitability	8
2.3.2	Performance Efficiency	9
2.3.3	Usability	9
2.3.4	Reliability	9
2.3.5	Maintainability	10
3	Specifica dei test	11
3.1	Analisi statica	11
3.2	Analisi dinamica	11
3.2.1	Test di unità [TU]	11
3.2.2	Test di integrazione [TI]	11
3.2.3	Test di sistema [TS]	12
3.2.4	Test di regressione [TR]	12
3.2.5	Test di accettazione [TA]	12
4	Metriche dei test	13
4.1	Metriche relative ai processi	13
4.1.1	<i>Schedule Variance (SV)_g</i>	13
4.1.2	<i>Budget Variance (BV)_g</i>	13
4.2	Metriche relative ai documenti	13
4.2.1	Numero di errori grammaticali	13
4.2.2	Gunning fog index	14
4.2.3	Indice di Gulpease	14
4.2.4	Indice di Flesh	14
4.3	Metriche relative ai prodotti software	15
4.3.1	Functional Implementation Completeness	15
4.3.2	Average Functional Implementation Correctness	15
4.3.3	Average Learning Time	16
4.3.4	Failure Density	16

4.4	Tabella riassuntiva delle metriche	17
5	Appendici	18
5.1	Resoconto delle attività di verifica	18
5.1.1	Riassunto delle attività di verifica	18
5.1.1.1	Revisione dei requisiti	18
5.1.2	Dettaglio delle verifiche tramite analisi	18
5.1.2.1	Processi	18
5.1.2.2	Documenti	19
5.1.3	Dettaglio dell'esito delle revisioni	19

1 Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo di esporre dettagliatamente le norme, le metodologie e gli standard che il gruppo 7DOS intende adottare per assicurare che ogni *prodotto_g*, di natura documentale o applicativa, aderisca ai vincoli di *qualità_g* stabiliti dal proponente. Per garantire il rispetto di tali vincoli si prevede un continuo *processo_g* di verifica delle attività svolte dal gruppo, al fine di individuare eventuali problematiche nel minor tempo possibile permettendo immediati interventi risolutivi.

1.2 Scopo del prodotto

Il prodotto da realizzare consiste in un *plug-in_g* per il software di monitoraggio *Grafana_g*, da sviluppare in linguaggio *JavaScript_g*. Il prodotto dovrà svolgere almeno le seguenti funzioni:

- Leggere la definizione di una *rete Bayesiana_g*, memorizzata in formato *JSON_g*;
- Associare dei nodi della rete Bayesiana ad un flusso di dati presente nel sistema di Grafana;
- Ricalcolare i valori delle probabilità della rete secondo regole temporali prestabilite;
- Derivare nuovi dati dai nodi della rete non collegati al flusso di dati, e fornirli al sistema di Grafana;
- Visualizzare i dati mediante il sistema di creazione di grafici e *dashboard_g* a disposizione.

Viene previsto un utilizzo del prodotto da parte dell'azienda proponente per il monitoraggio di sistemi gestionali in Cloud; tuttavia, dato l'obiettivo di rendere il prodotto open-source, esso dovrà essere utilizzabile indipendentemente dal particolare sistema che si desidera monitorare.

1.3 Glossario

Per rendere la lettura del documento più semplice, chiara e comprensibile viene allegato il *Glossario v1.0.0* nel quale sono contenute le definizioni dei termini tecnici, dei vocaboli ambigui, degli acronimi e delle abbreviazioni. La presenza di un termine all'interno del Glossario è segnalata con una "g" posta come pedice (esempio: *Glossario_g*).

1.4 Maturità del documento

Il presente documento sarà soggetto ad incrementi. Futuri per questo motivo, non si pone l'obiettivo di risultare completo già in questa fase del progetto. Tale decisione è dovuta al fatto che sono state trattate le esigenze di attività di progetto più impellenti e ricorrenti. Tutto ciò che riguarda la pianificazione degli incrementi, può essere trovato nel *Piano di Progetto V1.0.0* all'interno della quarta sezione.

1.5 Riferimenti

1.5.1 Normativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0.*;
- **Capitolato d'appalto C3:** G&B monitoraggio intelligente di processi DevOps
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C3.pdf>;
- **ISO/IEC 12207:**
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;

1.5.2 Informativi

- Slide dell'insegnamento Ingegneria del Software 2018-2019 - Verifica e validazione: analisi statica:
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L17.pdf>;
- Slide dell'insegnamento Ingegneria del Software 2018-2019 - Verifica e validazione: analisi dinamica:
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L18.pdf>;
- *Grafana_g* Code Styleguide:
<http://docs.grafana.org/plugins/developing/code-styleguide/>;
- *Angular TypeScript_g* Code Styleguide:
<https://angular.io/guide/styleguide>;
- **Verbali:** *Verbale del 2018-12-12*;
- **Software Engineering** - Ian Sommerville - 10th Edition.

2 Strategia generale per la verifica

Il Piano di Qualifica prevede che vengano delineati obiettivi da portare a termine seguendo strategia e metriche ben definite.

2.1 Obiettivi

La qualità di un qualsiasi prodotto o sistema è strettamente legata alla qualità dei processi che portano al suo sviluppo. Pertanto, nelle seguenti sezioni, verranno descritti gli obiettivi da perseguire nell'intento di garantire la massima qualità di processi e prodotti nella realizzazione del progetto.

2.2 Qualità di processo

È impossibile creare prodotti di alta qualità se il proprio *way of working_g* è scadente: risulta quindi fondamentale che i processi attuati, *in primis*, garantiscano un elevato livello qualitativo. Il gruppo 7DOS ha deciso di adottare la normativa *ISO/IEC 15504_g*, anche nota come *SPICE_g*, e di seguire il principio di miglioramento continuo (*PDCA_g*).

2.2.1 ISO/IEC 15504(SPICE)

Per ogni *processo_g* lo standard ISO/IEC 15504 definisce una scala di maturità a cinque livelli (più il livello base, detto "livello 0"), riportati di seguito:

- **Livello 0 - Incomplete Process:** il processo riporta *performance_g* e risultati incompleti, inoltre è gestito in modo caotico;
- **Livello 1 - Performed Process:** il processo raggiunge i risultati attesi ma viene eseguito in modo non controllato. Gli attributi di tale processo sono:
 - **1.1 - Process Performance:** capacità di un processo di raggiungere gli obiettivi definiti.
- **Livello 2 - Managed Process:** il processo è pianificato e tracciato secondo standard prefissati, dunque il suo prodotto è controllato, mantenuto e soddisfa determinati criteri di qualità. Gli attributi di tale processo sono:
 - **2.1 - Performance Management:** capacità di un processo di identificare gli obiettivi e di definire, monitorare e modificare le sue performance;
 - **2.2 - Work Product Management:** capacità di un processo di identificare, elaborare, documentare e controllare i propri risultati.
- **Livello 3 - Established Process:** il processo possiede specifici standard organizzativi che includono linee guida personalizzate, il tutto è consolidato tramite una politica di feedback del prodotto. Gli attributi di tale processo sono:

- **3.1 - Process Definition:** il processo specifico si basa su processi standard, individuando e incorporando caratteristiche fondamentali di questi;
- **3.2 - Process Deployment:** sono stati definiti ed assegnati dei ruoli a ciascun membro del team, ogni risorsa necessaria per l'esecuzione del processo è disponibile ed utilizzabile.
- **Livello 4 - Predictable Process:** il processo è quantitativamente misurato e statisticamente analizzato per permettere di prendere decisioni oggettive e per assicurare che le prestazioni rimangano all'interno di limiti definiti. Gli obiettivi sono, di conseguenza, supportati in maniera consistente. Gli attributi di tale processo sono:
 - **4.1 - Process Measurement:** i risultati di misurazione dei processi vengono utilizzati per garantire che le prestazioni del processo supportino il raggiungimento degli obiettivi determinati;
 - **4.2 - Process Control:** il processo è gestito quantitativamente in modo da renderlo stabile, capace e prevedibile entro limiti definiti.
- **Livello 5 - Optimizing Process:** il processo è in continuo miglioramento per raggiungere adeguatamente gli obiettivi prefissati. Gli attributi di tale processo sono:
 - **5.1 - Process Innovation:** gli obiettivi di miglioramento del processo supportano gli obiettivi aziendali rilevanti;
 - **5.2 - Process Optimization:** le modifiche alla definizione, gestione e le prestazioni del processo si traducono in un impatto efficace per il raggiungimento degli obiettivi.

Lo standard SPICE offre una scala di valutazione per ogni processo, in modo da misurare il livello di raggiungimento degli stessi:

- **N - Not Achieved:** 0 - 15%;
- **P - Partially Achieved:** >15% - 50%;
- **L - Largely Achieved:** >50% - 85%;
- **F - Fully Achieved:** >85% - 100%.

2.2.2 Ciclo di miglioramento continuo (PDCA)

Il ciclo di miglioramento continuo (PDCA, *Plan-Do-Check-Act*) prevede quattro fasi iterative che permettono di controllare costantemente lo sviluppo di un processo, in modo da poter perseguire la miglior qualità di quest'ultimo:

- **Plan:** in questa fase vengono definiti elementi estremamente importanti che riguardano il ciclo di miglioramento continuo. In particolare vengono fissati obiettivi, processi da utilizzare, risultati da ottenere, personale incaricato per i vari processi e scadenze da rispettare;

- **Do:** in questa fase vengono avviate tutte le attività previste da completare entro la data stabilita;
- **Check:** in questa fase vengono confrontati i risultati ottenuti dalle varie attività con quelli ipotizzati durante la fase Plan;
- **Act:** in questa fase vengono individuate le possibili problematiche che hanno prodotto risultati differenti da quelli attesi. Di conseguenza, verranno determinate tutte le attività da revisionare per migliorare la qualità del processo.

2.2.3 Gestione

Il gruppo 7DOS ha deciso di seguire l'approccio a maturità di processo per i seguenti motivi:

- predisposizione alla cura della qualità del prodotto e dei processi;
- previsto nelle buone pratiche di *management_g*;
- adatto per i neofiti della programmazione progettuale.

2.3 Qualità di prodotto

Per poter garantire che il prodotto realizzato sia di alta qualità, è necessario definire un modello per la valutazione di quest'ultima; il team 7DOS, per questo motivo, ha scelto di adottare il modello di qualità delineato nello standard *ISO/IEC 25010_g*), anche noto come *SQuaRE_g*.

Tale modello comprende 8 caratteristiche (ciascuna divisa in sotto-caratteristiche, per un totale di 31) che vanno prese in considerazione durante lo sviluppo del progetto per garantire un'elevata qualità complessiva del prodotto finale.

Per praticità e rilevanza ai fini del prodotto, sono state selezionate 5 caratteristiche da considerare e per ciascuna sono state individuate le sotto-caratteristiche più rilevanti al progetto, da perseguire come obiettivi prioritari. In particolare, sono state scartate: *Compatibility_g*, in quanto andando a realizzare un plug-in (di natura integrato in un sistema preesistente) è stata giudicata superflua; *Security_g*, in quanto il plug-in non dovrà gestire autenticazione o raccolta di dati; ed infine *Portability_g* in quanto essendo il prodotto un plug-in per un determinato sistema, non è rilevante la sua portabilità ad altri ambienti.

Di seguito vengono riportate le caratteristiche e sotto-caratteristiche selezionate.

2.3.1 Functional Suitability

Questa caratteristica esprime il grado di soddisfacimento dei requisiti espliciti ed impliciti da parte di un prodotto o servizio, quando utilizzato sotto determinate condizioni.

Sotto-caratteristiche notevoli:

- **Functional Completeness:** esprime il grado con cui l'insieme di funzioni copre i compiti specificati e gli obiettivi dell'utente;
- **Functional Correctness:** esprime il grado con cui il prodotto restituisce risultati corretti, entro il livello di precisione desiderato.

2.3.2 Performance Efficiency

Questa caratteristica esprime le prestazioni relative al sistema come, quantità di risorse utilizzate per eseguire una determinata funzionalità del sistema sotto specifiche condizioni.

Sotto-caratteristiche notevoli:

- **Time Behaviour:** esprime il grado con cui i tempi di risposta ed elaborazione e i volumi di produzione di un prodotto o sistema, durante l'esecuzione delle sue funzionalità, rispettano i requisiti;
- **Resource Utilization:** esprime il grado con cui il numero e tipo di risorse utilizzate da un prodotto o sistema, durante l'esecuzione delle sue funzionalità, rispetta i requisiti.

2.3.3 Usability

Questa caratteristica esprime il grado con cui un prodotto o sistema può essere usato da un determinato utente per raggiungere determinati scopi con efficacia, efficienza e soddisfazione in uno specifico contesto d'uso.

Sotto-caratteristiche notevoli:

- **Learnability:** esprime il grado con cui determinati utenti sono in grado di imparare ad utilizzare il prodotto o sistema efficacemente, efficientemente, con sicurezza da rischi e soddisfazione in un dato contesto d'uso;
- **Operability:** esprime il grado con cui un prodotto o sistema ha attributi che lo rendono facile da operare e controllare;
- **User Error Protection:** esprime il grado di efficacia ed efficienza con cui un sistema protegge gli utenti dal commettere errori.

2.3.4 Reliability

Questa caratteristica esprime il grado con cui un sistema, prodotto o componente esegue determinate funzioni sotto specifiche condizioni per un dato periodo di tempo.

Sotto-caratteristiche notevoli:

- **Maturity:** esprime il grado con cui un sistema, prodotto o componente raggiunge i requisiti di affidabilità in normali condizioni operative;

- **Fault Tolerance:** esprime il grado con cui un sistema, prodotto o componente opera come previsto nonostante la presenza di malfunzionamenti hardware o software.

2.3.5 Maintainability

Questa caratteristica esprime il grado di efficacia ed efficienza con cui un prodotto, o componente può essere modificato per migliorarlo, correggerlo o adattarlo a dei cambiamenti all'ambiente.

Sotto-caratteristiche notevoli

- **Modularity:** esprime il grado di scomposizione del sistema in parti minimali tali che un cambiamento ad una specifica componente ha il minimo impatto su tutte le altre componenti;
- **Analyzability:** esprime il grado di efficacia ed efficienza con cui è possibile analizzare l'impatto nel sistema di uno specifico cambiamento ad una o più delle sue parti, ai fini di rilevare eventuali casi di fallimento;
- **Modifiability:** esprime il grado con cui un prodotto o sistema può essere modificato efficacemente ed efficientemente senza introdurre difetti che ne possano intaccare la qualità complessiva;
- **Testability:** esprime il grado di efficacia ed efficienza con cui è possibile stabilire ed eseguire dei test per valutare la qualità del sistema.

3 Specifica dei test

3.1 Analisi statica

L'analisi statica è una tecnica di analisi applicabile sia ai documenti sia al codice che permette di individuare errori tramite verifica. Nella sezione 3.2 del documento *Norme di Progetto v1.0.0* vengono elencati gli strumenti per effettuare la verifica della documentazione. Per quanto riguarda il codice, attraverso l'analisi statica possono essere analizzate la sintassi e la semantica, stimate le variabili e controllato il flusso dei dati.

3.2 Analisi dinamica

L'analisi dinamica è una tecnica di analisi applicabile solamente al prodotto software. Essa permette di individuare errori di implementazione e di verificare il funzionamento del codice mediante l'uso di test dedicati durante l'esecuzione del codice stesso.

Essendo il test una parte essenziale per questo tipo di verifica, è necessario tutti i test siano *ripetibili*_g. Con ciò si intende che un test specifico eseguito in un ambiente definito produce sempre gli stessi output, dato un certo input.

I test applicabili sono di 5 tipi diversi, elencati nelle seguenti sottosezioni.

3.2.1 Test di unità [TU]

I test di unità mirano alla verifica della parte più piccola di lavoro prodotta da un programmatore, equivalente all'unità logica più piccola del prodotto, che può essere una singola classe, un metodo o funzione oppure un insieme di essi.

3.2.2 Test di integrazione [TI]

I test di integrazione mirano alla verifica di due o più unità già testate che vengono aggregate **incrementalmente** in una struttura più grande, rappresentando l'estensione logica del test di unità. In questo modo si può testare se il comportamento atteso dell'aggregato rispetta le previsioni. In caso negativo, è possibile che le singole unità contengano difetti residui da correggere oppure che i software utilizzati siano poco conosciuti e abbiano comportamenti inaspettati.

La strategia di integrazione dei vari moduli scelta è quella *bottom-up*_g, la quale prevede che vengano testate per prime le procedure più a basso livello e passando, progressivamente, a procedure di più alto livello.

3.2.3 Test di sistema [TS]

I test di sistema mirano alla verifica del prodotto software completo di tutte le sue componenti. Un software a cui vengono applicati questi tipi di test deve essere giunto ad una versione ritenuta definitiva.

Essendo test significativi, sarà richiesta la supervisione dei Verificatori incaricati.

3.2.4 Test di regressione [TR]

I test di regressione mirano ad eseguire nuovamente i test di unità e di integrazione su componenti software che hanno subito modifiche, in modo da controllare che i cambiamenti apportati non abbiano inserito nuovi errori sia nelle componenti modificate che nelle componenti non modificate e che prima non erano soggette ad errori.

3.2.5 Test di accettazione [TA]

I test di accettazione mirano al collaudo del prodotto software in presenza del *proponente*_g. Essi sono test finali e il superamento di essi comporta la validazione e il rilascio del prodotto.

4 Metriche dei test

Il processo di verifica, per essere informativo, deve essere quantificabile. Sono state quindi stabilite delle metriche di natura numerica, in modo da misurare accuratamente le varie caratteristiche di qualità del prodotto. Per ogni metrica sono presenti spiegazione, motivazione, formula usata per il calcolo e tre intervalli di valori. Questi ultimi sono distinti come segue:

- **Intervallo limite:** eventuali intervalli di valori in cui la metrica deve essere racchiusa per poter essere considerata come un valore legale che possiede una certa rilevanza;
- **Intervallo accettabile:** i valori che vengono considerati oltre la soglia di accettabilità per la metrica;
- **Intervallo obiettivo:** i valori che vengono considerati l'obiettivo ottimale da raggiungere.

4.1 Metriche relative ai processi

Le metriche per i processi scelte permettono di analizzare costi e tempi.

4.1.1 *Schedule Variance* $(SV)_g$

Metrica che indica se si è in linea, in anticipo o in ritardo rispetto alle attività pianificate nella *baseline_g*. Essa è un indicatore di efficacia.

Se $SV > 0$ significa che si sta procedendo più velocemente rispetto a quanto pianificato. Viceversa, se $SV < 0$ allora il gruppo è in ritardo.

4.1.2 *Budget Variance* $(BV)_g$

Metrica che indica se la spesa complessiva fino a quel momento è maggiore o minore rispetto a quanto pianificato. Essa è un indicatore con valore contabile e finanziario.

Se $BV > 0$ il budget complessivo sta diminuendo con velocità minore rispetto a quanto pianificato. Viceversa se $BV < 0$.

4.2 Metriche relative ai documenti

4.2.1 Numero di errori grammaticali

Misura del numero di errori grammaticali presenti all'interno di un documento. Tutti i documenti verranno analizzati da un apposito strumento di analisi grammaticale. Per poter essere accettati non potranno avere un numero di errori grammaticali superiore a zero.

4.2.2 Gunning fog index

Indice utilizzato per misurare la facilità di lettura e comprensione di un testo. Il numero risultante è un indicatore del numero di anni di educazione formale necessari al fine di leggere il testo con facilità.

L'indice di Gunning fog è calcolabile tramite la seguente formula:

$$0.4 * \left(\frac{n^{\circ} \text{ parole}}{n^{\circ} \text{ frasi}} \right) + 100 * \left(\frac{n^{\circ} \text{ parole complesse}}{n^{\circ} \text{ parole}} \right)$$

Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** $\mathbb{R}+$;
- **Intervallo accettabile:** $\mathbb{R}+$;
- **Intervallo obiettivo:** ≤ 12 .

4.2.3 Indice di Gulpease

Utilizzato per misurare la leggibilità di un testo in lingua italiana.

L'indice di Gulpease è calcolabile tramite la seguente formula:

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

I risultati sono compresi tra 0 e 100, dove 0 indica la leggibilità più bassa e 100 la leggibilità più alta. In generale risulta che i 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 la licenza superiore.

Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** 0-100;
- **Intervallo accettabile:** 40-100;
- **Intervallo obiettivo:** 60-100.

4.2.4 Indice di Flesh

Utilizzato per misurare la leggibilità di un testo in lingua inglese.

L'indice di Flesh è calcolabile tramite la seguente formula:

$$206,835 - (0,846 * S) - (1,015 * P)$$

Dove S è il numero di sillabe calcolato su un campione di 100 parole e P è il numero medio di parole per frase.

La leggibilità è considerata:

- **Alta:** se l'indice è superiore a 60;
- **Media:** se l'indice è compresa tra 50 e 60;
- **Bassa:** se l'indice è inferiore a 50.

Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** 0-100;
- **Intervallo accettabile:** 50-100;
- **Intervallo obiettivo:** 60-100.

4.3 Metriche relative ai prodotti software

Di seguito viene riportato solamente un piccolo sottoinsieme di metriche relative ai prodotti software ritenute come fondamentali dal team 7DOS. Questa sezione verrà ampliata successivamente qualora fosse necessario introdurre nuove metriche per la valutazione della qualità dei prodotti software.

4.3.1 Functional Implementation Completeness

Misurazione in percentuale del grado con cui le funzionalità offerte dalla corrente implementazione del software coprono l'insieme di funzioni specificate nei requisiti.

Questa metrica è stata scelta per valutare il grado di completezza del prodotto; l'obiettivo è implementare tutte le funzionalità richieste.

Viene utilizzata la seguente formula:

$$FI_{Comp} = \frac{NF_i}{NF_r} * 100$$

dove FI_{Comp} è il valore della metrica, NF_i è numero di funzioni attualmente implementate e NF_r è numero di funzioni specificate dai requisiti. Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** 0-100;
- **Intervallo accettabile:** 75-100;
- **Intervallo obiettivo:** 100.

4.3.2 Average Functional Implementation Correctness

Misurazione in percentuale del grado in cui le funzionalità offerte dalla corrente implementazione del software, in media, rispettano il livello di precisione indicato nei requisiti.

Questa metrica è stata scelta per valutare il grado di accuratezza e garantire la qualità dei risultati restituiti dal prodotto, in quanto andrà a fare previsioni sulla verosimiglianza di alcuni eventi in base ai dati forniti, ed è necessario che tali previsioni siano sufficientemente

accurate.

Viene utilizzata la seguente formula:

$$aFI_{Corr} = \frac{\sum_{i=1}^N \frac{iPF_i}{rPF_i}}{N} * 100$$

dove aFI_{Corr} è il valore della metrica, iPF_i è il livello di precisione della i -esima funzione implementata, rPF_i è il livello di precisione della i -esima funzione secondo i requisiti, e N è il numero totale di funzioni considerate. Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** 0-100;
- **Intervallo accettabile:** 80-100;
- **Intervallo obiettivo:** 95-100.

4.3.3 Average Learning Time

Misurazione in minuti del tempo medio impiegato da un utente per imparare ad utilizzare una singola funzionalità del prodotto.

Questa metrica è stata scelta poiché, trattandosi di un prodotto che verrà reso disponibile pubblicamente, è stato ritenuto importante renderlo semplice da imparare per permetterne l'uso ad una vasta gamma di utenti.

Viene utilizzata la seguente formula:

$$aLT = \frac{\sum_{i=1}^N LT_i}{N}$$

dove aLT è il valore della metrica, LT_i è il tempo necessario ad imparare ad utilizzare la i -esima funzione implementata, espresso in minuti, e N è il numero totale di funzioni considerate. Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** \mathbb{R}_+ ;
- **Intervallo accettabile:** 0-30;
- **Intervallo obiettivo:** 0-15.

4.3.4 Failure Density

Misurazione in percentuale della quantità di test falliti rispetto alla quantità di test eseguiti. Questa metrica è stata scelta per garantire che il prodotto sia generalmente stabile e non risulti poco utilizzabile o inutilizzabile a causa di eccessive *failure*_g. Valutazioni più precise saranno effettuate in base ai singoli risultati dei test.

Viene utilizzata la seguente formula:

$$FD = \frac{T_f}{T_c} * 100$$

dove FD è il valore della metrica, T_f è il numero di test falliti e T_e è il numero di test eseguiti. Per questa metrica vengono definiti i seguenti intervalli:

- **Intervallo limite:** 0-100;
- **Intervallo accettabile:** 0-10;
- **Intervallo obiettivo:** 0.

4.4 Tabella riassuntiva delle metriche

Nome Metrica	Intervallo limite	Range accettabile	Range ottimale
Functional Implementation Completeness	0-100	75-100	100
Average Functional Implementation Correctness	0-100	80-100	95-100
Numero di errori grammaticali	/	0	0
Gunning fog index	/	12-15	0-12
Indice di Gulpease	0-100	40-100	60-100
Indice di Flesh	0-100	50-100	60-100
Average Learning Time	/	0-15	0-30
Failure Density	0-100	0-10	0

5 Appendici

5.1 Resoconto delle attività di verifica

5.1.1 Riassunto delle attività di verifica

5.1.1.1 Revisione dei requisiti

Durante il periodo successivo alla fine della stesura di ogni documento, i *Verificatori* hanno provveduto all'attività di verifica. Quest'attività è stata eseguita seguendo le linee guida contenute nelle *Norme di Progetto v1.0.0* sezione 3.2.2. Si sono poi calcolate per i documenti le metriche descritte nel punto 2.2.2. Infine sono stati verificati i processi seguendo le specifiche contenute nelle *Norme di Progetto v1.0.0* sezione 3.2.1 e vi sono state calcolate le metriche contenute nel punto 2.2.1 contenute in questo documento.

5.1.2 Dettaglio delle verifiche tramite analisi

5.1.2.1 Processi Il grafico PDCA della fase di Analisi è:

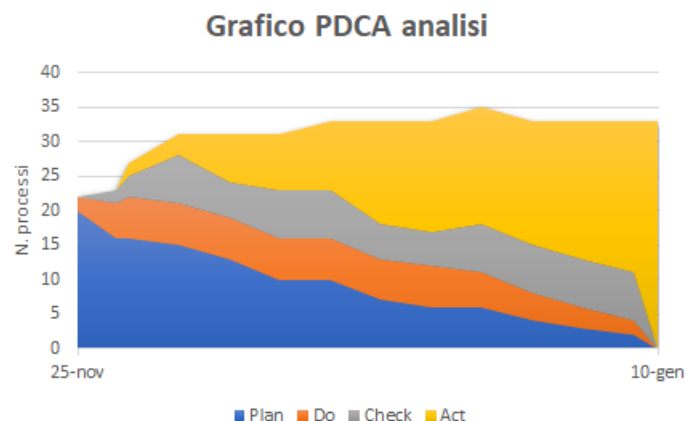


Figura 1: Grafico PDCA, fase di Analisi

Dal grafico possiamo estrapolare che:

- Si è notato alcuni mutamenti dei processi pianificati, dovuti ad errori di pianificazione dati dalla poca esperienza del gruppo di lavoro;
- Si può notare come il gruppo abbia cercato di rendere omogenea nel tempo l'avanzamento dei processi, alcuni rallentamenti sono dovuti alla sovrapposizione degli impegni universitari dei componenti del gruppo con la realizzazione del progetto. Nel complesso si vede come l'omogeneità è stata abbastanza rispettata.

5.1.2.2 Documenti

Vengono qui riportati i valori dell'indice Gulpease per ogni documento durante la fase di Analisi. Un documento è considerato valido soltanto se rispetta le metriche descritte su 4.1.

Documento	Valore indice	Esito
<i>Piano di Progetto v1.0.0</i>	55.49	Superato
<i>Norme di Progetto v1.0.0</i>	54.54	Superato
<i>Analisi dei Requisiti v1.0.0</i>	58.80	Superato
<i>Piano di Qualifica v1.0.0</i>	53.98	Superato
<i>Studio di Fattibilità v1.0.0</i>	50.77	Superato
<i>Glossario v1.0.0</i>	51.66	Superato

Tabella 1: Esito verifica documenti

Dalla tabella si può notare come tutti gli indici Gulpease dei documenti rientrino nei vincoli dati. Per questo motivo i documenti redatti hanno raggiunto la leggibilità desiderata.

5.1.3 Dettaglio dell'esito delle revisioni

Le revisioni del committente previste durante lo sviluppo del progetto sono quattro. Al termine di ogni revisione, il committente segnalerà le problematiche riscontrate attraverso una valutazione globale dell'andamento del progetto ed una dettagliata per ciascun documento. Questo aiuterà il gruppo a eliminare problemi e criticità nel progetto per poi procedere su una base verificata e il più possibile corretta.