



Progetto: *HD Viz*
codebusterswe@gmail.com

Piano Di Qualifica

Informazioni sul documento	
Versione	1.0.0
Approvatori	Baldisseri Michele
Redattori	Scialpi Paolo Zenere Marco Rago Alessandro Pirolo Alessandro
Verificatori	Baldisseri Michele Scialpi Paolo
Uso	Esterno
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo Gruppo <i>CodeBusters</i>

Descrizione

Documento che descrive le operazione di validazione e verifica seguite durante il progetto *HD Viz*.

Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
0.0.3	30-12-2020	Zenere Marco	Analista	Stesura qualità di prodotto
0.0.2	29-12-2020	Rago Alessandro	Analista	Stesura §4
0.0.1	21-12-2020	Zenere Marco	Analista	Creazione bozza documento

Indice

Elenco delle tabelle

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di fornire tutte le informazioni relative al sistema di controllo di qualità per i processi ed i prodotti, basandosi su assunti misurabili ma adattati alle esigenze del proprio progetto. Esso deve implementare degli standard che permettano il miglioramento continuo, tracciando periodicamente tramite misurazioni i risultati ottenuti sfruttandoli per definire azioni migliorative. All'interno del **Piano di Qualifica** vengono anche raccolte le definizioni dei test, il loro stato e il loro tracciamento.

1.2 Scopo del capitolato

Oggigiorno, anche i programmi più tradizionali gestiscono e memorizzano una grande mole di dati e di conseguenza serve un software in grado di eseguire un'analisi e una interpretazione delle informazioni.

Il capitolato^G C4 ha come obiettivo quello di creare un'applicazione di visualizzazione di dati con numerose dimensioni in un formato comprensibile dall'occhio umano. A questo scopo è necessario utilizzare algoritmi di intelligenza artificiale, o nel caso svilupparne di nuovi, che, agendo sulla distanza dei vari punti del grafico, riescano a sviluppare un modello semplificato che ne evidenzi i cluster^G. L'applicazione dovrà inoltre agire su questi grafici creati evidenziando i dati ottenuti.

1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzare, è stato compilato il *Glossario 1.0.0*. In questo documento sono riportati tutti i termini di particolare importanza e con un significato particolare. Questi termini sono evidenziati da una 'G' ad apice.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- Capitolato d'appalto C4 - HD Viz: visualizzazione di dati multidimensionali:
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C4.pdf>

1.4.2 Riferimenti informativi

- Norme di Progetto v1.0.0;
- Standard ISO/IEC 12207:1995:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf

- Guide to the Software Engineering Body of Knowledge(SWEBOK), 2014
- Software Engineering - Ian Sommerville - 10 th Edition (2010):
(formato cartaceo);
- Slide T3 del corso Ingegneria del Software - Ciclo di vita del software:
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L03.pdf>

2 Qualità di processo

2.1 Introduzione

Per garantire la qualità dei processi è stato utilizzato come riferimento lo standard ISO/IEC/IEEE 12207:1995. Tra i processi elencati dal modello, il gruppo ne ha scelti alcuni che sono stati semplificati e adattati alle necessità del progetto. Questa sezione espone i valori di qualità accettabili sulla base di metriche elencate nelle *Norme di Progetto v 1.0.0*. Di seguito sono esposti i processi selezionati.

2.2 Processi Primari

2.2.1 Analisi dei Requisiti

L'analisi dei requisiti si occupa di trasformare informazioni raccolte dalle fonti diagrammi di casi d'uso e requisiti. Questo documento descrive il funzionamento generale del prodotto software e di ogni sua parte.

2.2.1.1 Metriche

- MPC3 - Requirements stability index (RSI);
- MPC4 - Satisfied obligatory requirements (SOR);

2.2.1.2 Valori ammissibili

Metrica	Valori accettabile	Valore ottimale
MPC3		
MPC4	100%	100%

2.2.2 Progettazione di dettaglio

2.3 Processi di supporto

2.3.1 Pianificazione

2.3.2 Verifica

2.3.3 Documentazione

2.4 Processi Organizzativi

2.4.1 Gestione della qualità

3 Qualità di prodotto

Dallo standard ISO/IEC 9126, il gruppo *CodeBusters* ha identificato le qualità che ritiene necessarie nell'intero ciclo di vita del prodotto e ne ha tratto delle metriche e degli obiettivi da realizzare per perseguire la qualità del software.

3.1 Prodotti

3.1.1 Documenti

I documenti dovranno essere comprensibili e corretti ortograficamente e sintatticamente.

3.1.1.1 Comprensione

I documenti dovranno poter essere letti e compresi da tutti coloro che hanno una istruzione di base ovvero che abbiano completato la scuola secondaria di secondo grado.

3.1.1.2 Correttezza

La correttezza dei documenti sarà compito del verificatore con l'ausilio di uno strumento di controllo ortografico.

3.1.1.3 Metriche utilizzate

- **MPD1 Indice Gulpease:** questo indice descrive la leggibilità del documento prodotto;
- **MPD2 Errori ortografici:** descrive la correttezza ortografica del documento.

3.1.2 Software

3.1.2.1 Funzionalità

Capacità del prodotto di offrire tutte le funzioni individuate nell'Analisi dei Requisiti. Gli obiettivi da perseguire sono:

- **Accuratezza:** il prodotto dovrà ottenere i risultati richiesti;
- **Adeguatezza:** le funzionalità dovranno almeno equivalere le attese.

3.1.2.1.1 Metrica utilizzata

MPD3 Copertura dei requisiti: descrive quanti requisiti sono stati implementati nel software;

3.1.2.2 Affidabilità

Con questo termine si intende la capacità del prodotto di riuscire a svolgere tutte le funzionalità presenti anche in caso di errori o problemi. L'esecuzione, per risultare affidabile, dovrà possedere queste caratteristiche:

- **Tolleranza agli errori:** la gestione degli errori dovrà essere tale da permettere di avere sempre un alto livello di prestazioni;
- **Previdenza:** evitare che malfunzionamenti o operazioni illegali si manifestino.

3.1.2.3 Usabilità

Il prodotto dovrà essere comprensibile e graficamente armonioso in modo da rendere piacevole l'esperienza dell'utente. Gli obiettivi di usabilità sono:

- **Comprensibilità:** chi utilizza il prodotto deve comprendere facilmente quali sono le sue funzionalità in modo da ottenere i risultati voluti;
- **Facilità d'uso:** l'utente deve imparare senza troppe difficoltà come utilizzare l'applicazione;
- **Operabilità:** le funzioni devono essere compatibili con le aspettative dell'utente;
- **Attrattiva:** il software deve essere piacevole all'occhio.

3.1.2.3.1 Metrica utilizzata

MPD6 Average Cyclomatic Complexity: indica il numero di cammini indipendenti presenti nel programma.

3.1.2.4 Efficienza

L'efficienza è la capacità di raggiungere un fine con il minor utilizzo di tempo e risorse. Per quanto riguarda il prodotto questo dovrà essere:

- **Veloce:** le risposte all'input dell'utente devono essere quanto più possibili veloci e corrette;
- **Leggero:** il software dovrà utilizzare meno risorse possibili dell'utente.

3.1.2.5 Manutenibilità

Un prodotto, per avere tale capacità, dovrà permettere future correzioni e modifiche senza che ciò rischi di compromettere l'intero progetto. Le caratteristiche che il software deve avere sono:

- **Analizzabilità:** l'individuazione degli errori deve essere facile;
- **Modificabilità:** la modifica o l'aggiunta di nuove parti deve essere permessa. Il codice deve essere leggibile così da poterla inserire facilmente;

- **Stabilità:** dopo la modifica non devono insorgere altri problemi relativi alla incompatibilità con altre parti di codice;
- **Testabilità:** i test sulle modifiche effettuate devono essere facilmente implementati.

3.1.2.5.1 Metriche utilizzate

- **MPD4 Code Coverage:** descrive quanto il codice prodotto è coperto dalla suite di test dinamici;
- **MPD5 :** indica il numero di linee di codice in un file.

3.1.2.6 Portabilità

È la capacità di poter funzionare in diversi ambienti. Per avere tale capacità il prodotto dovrà avere le seguenti caratteristiche:

- **Adattabilità:** il software dovrà poter essere eseguito in numerosi browser senza che debbano essere effettuate delle modifiche;
- **Sostituibilità:** il software deve poter sostituire un prodotto con lo stesso fine e che viene eseguito sullo stesso browser.

Codice	Nome	Range Accettazione	Range Ottimalità
MPD1	Indice Gulpease	70-100%	90-100%
MPD2	Errori ortografici (EO)	100% corretto	100% corretto
MPD3	Copertura dei requisiti (CDR)	100% dei requisiti obbligatori	100% dei requisiti obbligatori
MPD4	Code Coverage (CC)	100%	100%
MPD5	Source Line of Code (SLOC)	da definire	da definire
MPD6	Average Cyclomatic complexity (ACC)	15	10

Tabella 2: Tabella delle metriche

4 Specifica dei test

I test costituiscono il cuore delle attività di analisi dinamica sul codice. Come già riportato nelle *Norme di Progetto* questi possono essere di vari tipi, ognuno con lo scopo di individuare diversi difetti software.

Il gruppo *CodeBusters* ha deciso che, per perseguire correttezza continua del prodotto software, il processo di verifica si svolgerà in parallelo con quello di sviluppo (Modello a V^G), facendo sì che questo non rallenti in alcun modo la produzione. L'obiettivo è quindi quello di rendere i test il più automatici possibili con l'utilizzo di software appositi.

4.1 Test di unità

I test di unità mirano a trovare errori verificando singoli metodi, singole classi, indipendentemente dal sistema che le circonda. A questo scopo è fondamentale che la dipendenza fra le classi sviluppate sia il più possibile "sana", ossia che lega due componenti perché essi vivono nello stesso contesto e devono essere testati insieme. In questa prima versione del *Piano di Qualifica* il gruppo non è in grado di stabilire dei test di unità funzionali, non avendo ancora iniziato il processo di progettazione e sviluppo software.

4.2 Test di integrazione

I test di integrazione verificano come le componenti software si integrino fra di loro. In questa prima versione del *Piano di Qualifica* il gruppo non è in grado di stabilire dei test di integrazione, non avendo individuato e testato le componenti del prodotto software.

4.3 Test di sistema

In base ai requisiti trovati e riportati nell'*Analisi dei Requisiti*, il gruppo ha stilato una lista di test di sistema da effettuare per verificarne la correttezza. Il codice utilizzato per l'identificazione dei test è specificato nelle Norme di Progetto, mentre delle sigle utili per comprendere la tabella seguente sono:

- **I**: test implementato;
- **NI**: test non implementato.

Codice	Descrizione	Stato
TS1F1	<p>L'utente deve poter caricare dei dati nel sistema tramite file CSV o interrogazione al DB. Se tali operazioni non sono eseguite correttamente il caricamento fallisce. Verificare che l'utente possa:</p> <ul style="list-style-type: none"> • Visualizzare correttamente la schermata di inserimento dei dati; • Inserire i dati attraverso file CSV tramite apposito bottone; • Il file CSV inserito sia sintatticamente corretto; • Inserire i dati attraverso una interrogazione al DB tramite query; • L'interrogazione sia stata effettuata correttamente; • Sia visualizzato a schermo un messaggio d'esito dell'operazione; • In caso di errore sia possibile reinserire i dati tramite file CSV o interrogazione al DB dalla stessa schermata. 	NI
TS2F2	<p>Aiuti all'utente attraverso widget. Verificare che:</p> <ul style="list-style-type: none"> • Siano tutti facilmente localizzabili dall'utente; • Siano tutti facilmente utilizzabili dall'utente; • Siano effettivamente utili per l'utente. 	NI

Continua nella pagina successiva...

L'applicazione deve fornire diverse visualizzazioni e diversi ridimensionamenti dei dati. Verificare che:

TS1F3

- Siano mostrati a video dei radio button per la selezione del grafico da utilizzare;
- Siano mostrati a video dei bottoni per la selezione delle dimensioni da visualizzare;
- Tutti i bottoni abbiano nomi parlanti;
- Ogni bottone sia selezionabile e permetta la visualizzazione del grafico (eventualmente ridimensionato) corretto.

NI

L'applicazione deve fornire la visualizzazione Scatter plot Matrix^G. Verificare che:

TS1F3.2

- Gli scatter plot creati siano disposti a matrice;
- Ogni scatter plot sia corretto rispetto ai dati da rappresentare;
- Ogni scatter plot metta in relazione variabile diverse;
- Ogni variabile sia in relazione con tutte le altre nei diversi scatter plot;
- L'utente possa selezionare quali scatter plot evidenziare perché più rilevanti.

NI

L'applicazione deve fornire la visualizzazione Heat Map^G e permettere di riordinare i punti. Verificare che:

TS1F3.3

- Ogni variabile sia in relazione con tutte le altre;
- I colori utilizzati rappresentano effettivamente la distanza tra i punti da misurare;
- Sia presente un dendrogramma^G ai bordi della mappa;
- L'utente possa ordinare i punti attraverso parametri specifici;
- Il grafico cambi aspetto in real time una volta cambiato parametro d'ordinamento;

NI

Continua nella pagina successiva...

TS1F3.4	L'applicazione deve fornire la visualizzazione Force Field ^G . Verificare che:	
	<ul style="list-style-type: none"> • Ogni variabile abbia il suo corrispettivo nodo nel grafico; • Ogni relazione sia identificata da una linea di collegamento tra i nodi; • Le distanze nello spazio a molte dimensioni siano correttamente tradotte in forze di attrazione e repulsione tra i nodi; • Il grafico sia bidimensionale (o tridimensionale). 	NI

TS3F3.4.1	L'utente deve poter utilizzare funzioni di "forza" diverse da quelle previste in automatico dal grafico "forcebased" di D3.js ^G e funzioni di calcolo della distanza diverse da quella standard Euclidea. Verificare che:	
	<ul style="list-style-type: none"> • L'utente possa scegliere la funzione di forza da utilizzare per la riduzione dimensionale da usare nel grafico Force Field^G; • L'utente possa scegliere la funzione di calcolo della distanza da utilizzare per la riduzione dimensionale in ogni grafico che la utilizza; • Il grafico cambi d'aspetto in real time. 	NI

TS1F3.5	L'applicazione deve fornire la visualizzazione Proiezione Lineare Multi Asse ^G . Verificare che:	
	<ul style="list-style-type: none"> • I punti dello spazio multidimensionale siano correttamente posizionati nel piano cartesiano; • La riduzione a due dimensioni sia corretta; • L'utente possa spostare gli assi per individuare le strutture di dati di suo interesse; • Lo spostamento degli assi avvenga spostando manualmente le frecce degli assi. 	NI

Continua nella pagina successiva...

L'applicazione deve fornire altre visualizzazioni con più di tre dimensioni. Verificare che:

- | | | |
|----------------|---|----|
| TS3F3.6 | <ul style="list-style-type: none">• L'utente possa selezionare un grafico a più di tre dimensioni;• IL grafico selezionato sia visualizzato correttamente. | NI |
|----------------|---|----|

Implementare analisi automatiche per evidenziare situazioni di particolare interesse. Verificare che:

- | | | |
|--------------|---|----|
| TS3F4 | <ul style="list-style-type: none">• In ogni tipo di grafico sia possibile la visualizzazione dei dati rilevanti;• Vengano effettivamente esclusi tutti i dati outlier;• Il variare delle dimensioni non modifichi i dati rilevanti. | NI |
|--------------|---|----|

Utilizzo di algoritmi di preparazione del dato per la visualizzazione. Verificare che:

- | | | |
|--------------|---|----|
| TS3F5 | <ul style="list-style-type: none">• La trasformazione dei dati avvenga correttamente;• La trasformazione adottata sia utile alla visualizzazione dei dati. | NI |
|--------------|---|----|

Presenza di una guida introduttiva per l'utente. Verificare che:

- | | | |
|--------------|---|----|
| TS3F6 | <ul style="list-style-type: none">• L'utente possa facilmente trovare e consultare la guida;• La guida sia scritta in italiano corretto;• La guida sia fruibile da ogni tipo di utente, anche quello meno esperto;• La guida spieghi tutti gli utilizzi dell'applicazione. | NI |
|--------------|---|----|

Continua nella pagina successiva...

Possibilità di visualizzare contemporaneamente due grafici per confronti. Verificare che:

TS2F7	<ul style="list-style-type: none">• Sia presente un bottone che permetta la visualizzazione di un secondo grafico;	NI
	<ul style="list-style-type: none">• L'utente possa effettivamente selezionare due grafici diversi nel radio button;	
	<ul style="list-style-type: none">• Il secondo grafico corrisponda a quello selezionato;	
	<ul style="list-style-type: none">• I dati rilevanti siano evidenziati correttamente anche nel secondo grafico.	

L'utente può salvare la sessione in corso per ripristinarla in un secondo momento. Verificare che:

TS2F8	<ul style="list-style-type: none">• Sia disponibile un bottone per il salvataggio della sessione;	NI
	<ul style="list-style-type: none">• Sia disponibile un bottone per il ripristino della sessione;	
	<ul style="list-style-type: none">• La sessione sia salvata correttamente;	
	<ul style="list-style-type: none">• La sessione sia effettivamente ripristinabile;	

Tabella 3: Test di sistema

Come si può notare lo stato di tutti i test è NI ("non implementato"). L'implementazione dei test e i loro risultati saranno oggetto della prossima versione del *Piano di Qualifica*.

CodiceTest	CodiciRequisiti
TS1F1	R1F1, R1F1.1, R1F1.2, R1F1.3
TS2F2	R2F2
TS1F3	R1F3, R1F3.1
TS1F3.2	R1F3.2
TS1F3.3	R1F3.3
TS1F3.4	R1F3.4

Continua nella pagina successiva...

TS1F3.4.1	R1F3.4.1, R1F3.7
TS1F3.5	R1F3.5
TS1F3.6	R1F3.6
TS3F4	R1F4
TS3F5	R1F5
TS3F6	R1F6
TS3F7	R1F7
TS3F8	R1F8

Tabella 4: Tracciamento test - requisiti funzionali

4.4 Test di accettazione

I test di accettazione rappresentano il collaudo del prodotto software, verificando che corrisponda con quello atteso dal proponente. Sono l'unione dei test di sistema già svolti dal gruppo durante lo sviluppo ed ulteriori test finali. In questa prima versione del *Piano di Qualifica* il gruppo non è in grado di stilare dei test aggiuntivi oltre quelli di sistema già riportati nella tabella ??.

4.5 Test di regressione

I test di regressione permettono di individuare errori causati da modifiche apportate a nuove versioni del prodotto. In questa prima versione del *Piano di Qualifica* il gruppo non è in grado di stabilire dei test di regressione utili.

5 Standard di qualità

5.1 ISO/IEC 9126

5.1.1 Metriche per la qualità interna

5.1.2 Metriche per la qualità esterna

5.1.3 Metriche per la qualità in uso

5.1.4 Metriche della qualità del software

6 Resoconto attività di verifica

6.1 Revisione dei requisiti

6.1.1 Tracciamento dei casi d'uso e dei requisiti

6.1.2 Analisi statica dei documenti

6.1.3 Esiti verifiche automatizzate

7 Valutazioni per il miglioramento

7.0.1 Valutazioni sull'organizzazione

7.0.2 Valutazioni sui ruoli

7.0.3 Valutazioni sugli strumenti di lavoro