



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
0.0.2	29-12-2020	Rago Alessandro	Analista	Stesura §4
0.0.1	21-12-2020	Zenere Marco	Analista	Creazione bozza documento

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del capitolato	5
1.3	Glossario	5
1.4	Riferimenti	5
1.4.1	Riferimenti normativi	5
1.4.2	Riferimenti informativi	5
2	Qualità di processo	7
2.1	Introduzione	7
2.2	Processi Primari	7
2.2.1	primo processo	7
2.2.1.1	Metriche	7
2.2.1.2	Valori ammissibili	7
2.2.2	Progettazione di dettaglio	7
2.3	Processi di supporto	8
2.3.1	Pianificazione	8
2.3.2	Verifica	8
2.3.3	Documentazione	8
2.4	Processi Organizzativi	9
2.4.1	Gestione della qualità	9
3	Qualità di prodotto	10
3.1	Prodotti	10
3.1.1	Documenti	10
3.1.1.1	Comprensione	10
3.1.1.2	Correttezza	10
3.1.1.3	Metrica utilizzata	10
3.1.2	Software	10
3.1.2.1	Funzionalità	10
3.1.2.2	Affidabilità	10
3.1.2.3	Usabilità	11
3.1.2.4	Efficienza	11
3.1.2.5	Manutenibilità	11
3.1.2.6	Portabilità	11
4	Specifica dei test	12
4.1	Test di unità	12
4.2	Test di integrazione	12
4.3	Test di sistema	12
4.4	Test di accettazione	19

4.5	Test di regressione	19
5	Standard di qualità	20
5.1	ISO/IEC 9126	20
5.1.1	Metriche per la qualità interna	20
5.1.2	Metriche per la qualità esterna	20
5.1.3	Metriche per la qualità in uso	20
5.1.4	Metriche della qualità del software	20
6	Resoconto attività di verifica	21
6.1	Revisione dei requisiti	21
6.1.1	Tracciamento dei casi d'uso e dei requisiti	21
6.1.2	Analisi statica dei documenti	21
6.1.3	Esiti verifiche automatizzate	21
7	Valutazioni per il miglioramento	22
7.0.1	Valutazioni sull'organizzazione	22
7.0.2	Valutazioni sui ruoli	22
7.0.3	Valutazioni sugli strumenti di lavoro	22

Elenco delle tabelle

2 Test di sistema 19

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 evidenzia 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 primo processo

2.2.1.1 Metriche

2.2.1.2 Valori ammissibili

Metrica	Valori accettabile	Valore ottimale
---------	-----------------------	--------------------

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 Metrica utilizzata

MPD1 Indice Gulpease: questo indice descrive la leggibilità del documento prodotto e affinché venga accettato il valore deve risiedere tra 50-100.

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.

3.1.2.2 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.

3.1.2.4 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.

3.1.2.6 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.

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. 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;• Inserire i dati attraverso una interrogazione al DB tramite query.	NI
TS1F1.1	<p>Caricamento dati attraverso l'invio di un file CSV. Verificare che:</p> <ul style="list-style-type: none">• Il file CSV inserito sia sintatticamente corretto;• I dati siano correttamente prelevati dal file e pronti per la visualizzazione;• Sia visualizzato a schermo un messaggio di corretto inserimento dei dati.	NI
TS1F1.2	<p>Caricamento dati attraverso l'interrogazione a un DB. Verificare che:</p> <ul style="list-style-type: none">• L'interrogazione sia stata effettuata correttamente;• I dati siano stati individuati e ritornati;• Sia visualizzato a schermo un messaggio che notifica che sono stati trovati i dati.	NI

Continua nella pagina successiva...

TS1F1.3	<p>Il caricamento fallisce se il file CSV non è corretto o l'interrogazione al DB fallisce. Verificare che:</p> <ul style="list-style-type: none">• Sia visualizzato a schermo un messaggio d'errore parlante;• 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
TS1F3	<p>L'applicazione deve fornire diverse visualizzazioni per i dati. Verificare che:</p> <ul style="list-style-type: none">• Siano mostrati a video dei radio button per la selezione del grafico da utilizzare;• I radio button abbiano nomi parlanti;• Ogni radio button sia selezionabile e permetta la visualizzazione del grafico corretto;• Sia possibile selezionare due grafici diversi nel caso in cui si voglia compararli.	NI

Continua nella pagina successiva...

TS1F3.1

L'applicazione deve permettere la scelta delle dimensioni da visualizzare. Verificare che:

- Siano mostrati a video dei bottoni per la selezione delle dimensioni da visualizzare;
- I bottoni devono avere nomi parlanti;
- Coerenza tra dimensioni visualizzabili e grafico scelto;
- Ogni bottone deve ridurre/aumentare le dimensioni correttamente rispetto al grafico visualizzato.

NI

TS1F3.2

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

- 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

TS1F3.3

L'applicazione deve fornire la visualizzazione Heat Map^G. Verificare che:

- 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.

NI

Continua nella pagina successiva...

TS1F3.3.1	<p>L'applicazione deve ordinare i punti nella visualizzazione Heat Map^G. Verificare che:</p> <ul style="list-style-type: none">• L'utente possa ordinare i punti attraverso parametri specifici;• Il parametro d'ordinamento sia selezionabile da un rudio button;• Il grafico effettivamente cambi aspetto in real time;• Il valore dei dati non cambi, ma solo la loro visualizzazione;	NI
TS1F3.4	<p>L'applicazione deve fornire la visualizzazione Force Field^G. Verificare che:</p> <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	<p>Utilizzo di funzioni di "forza" diverse da quelle previste in automatico dal grafico "forcebased" di D3.js^G. Verificare che:</p> <ul style="list-style-type: none">•••	NI

Continua nella pagina successiva...

TS1F3.5	<p>L'applicazione deve fornire la visualizzazione Proiezione Lineare Multi Asse^G. Verificare che:</p> <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
TS3F3.6	<p>L'applicazione deve fornire altre visualizzazioni con più di tre dimensioni. Verificare che:</p> <ul style="list-style-type: none"> • • • 	NI
TS3F3.7	<p>Utilizzo di funzioni di calcolo della distanza diverse dalla distanza "Euclidea" in tutte le visualizzazioni che dipendono da tale concetto. Verificare che:</p> <ul style="list-style-type: none"> • • • 	NI
TS3F4	<p>Implementare analisi automatiche per evidenziare situazioni di particolare interesse. Verificare che:</p> <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

Continua nella pagina successiva...

TS3F5

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

-
-
-

NI

TS3F6

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

- L'utente possa facilmente trovare e consultare la guida;
- La guida sia scritta un italiano corretto;
- La guida sia fruibile da ogni tipo di utente, anche quello meno esperto;
- La guida spieghi tutti gli utilizzi dell'applicazione.

NI

TS2F7

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

- Sia presente un bottone che permetta la visualizzazione di un secondo grafico;
- L'utente possa effettivamente selezionare due grafici diversi nel radio button;
- Il secondo grafico corrisponda a quello selezionato;
- I dati rilevanti siano evidenziati correttamente anche nel secondo grafico.

NI

Continua nella pagina successiva...

TS2F8	<p>L'utente può salvare la sessione in corso per ripristinarla in un secondo momento. Verificare che:</p> <ul style="list-style-type: none">• Sia disponibile un bottone per il salvataggio della sessione;• Sia disponibile un bottone per il ripristino della sessione;• La sessione sia salvata correttamente;• La sessione sia effettivamente ripristinabile;	NI
-------	--	----

Tabella 2: Test di sistema

Come si può notare lo stato di tutti i test è NI ("non implementato") visto che il gruppo non ha ancora iniziato la scrittura del codice. L'implementazione dei test e i loro risultati saranno oggetto della prossima versione del *Piano di Qualifica*.

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 2.

4.5 Test di regressione

I test di regressione permettono di individuare errori causati da modifiche apportate a nuove versioni del prodotto. Tali modifiche potrebbero infatti causare difetti in parti di programma già testate e corrette in versioni precedenti. 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