



Il gruppo Avant-Garde

sweavantgarde@gmail.com

PIANO DI QUALIFICA

Informazioni sul documento:

Versione	1.0.0
Approvazione	Jessica Carretta
Redazione	Lorenzo Pasqualotto
Verifica	Andrea Mangolini
Uso	Esterno

Registro delle Modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
v0.0.1	27-12-23	Lorenzo Pasqualotto	Amministratore	Stesura scheletro del documento
v0.0.2	29-12-23	Lorenzo Pasqualotto	Amministratore	Stesura sezione relativa a Qualità di prodotto
v0.0.3	05-01-24	Lorenzo Pasqualotto	Amministratore	Stesura sezione relativa a Qualità di processo
v0.0.4	06-01-24	Lorenzo Pasqualotto	Amministratore	Stesura sezione Testing
v0.1.0	20-01-24	Andrea Mangolini	Verificatore	Verifica dei contenuti dalla versione v0.0.1 alla versione v0.0.4
v1.0.0	12-02-24	Jessica Carretta	Responsabile	Approvazione del documento alla versione v1.0.0

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Il progetto	4
2	Metriche	5
2.1	Codice	5
2.2	Contenuto	5
3	Qualità di processo	6
3.1	Processi primari	6
3.1.1	Fornitura	6
3.1.2	Sviluppo	7
3.1.2.1	Progettazione	7
3.1.2.2	Codifica	8
3.2	Processi di supporto	9
3.2.1	Documentazione	9
3.2.2	Accertamento della qualità	9
3.2.3	Verifica	10
4	Qualità di prodotto	11
4.1	Customer needs	11
4.2	Efficienza	11
4.3	Usabilità	11
4.4	Portabilità su altre piattaforme	12
4.5	Manutenibilità	13
5	Test	14
5.1	Test di Unità	14
5.2	Test di Integrazione	14
5.3	Test di Sistema	14
5.4	Test di Regressione	14
5.5	Test di Accettazione	14
6	Resoconto	15
7	Riferimenti esterni	16

Note

Si tenga presente che alcuni termini utilizzati nel documento riportano la lettera **G** in apice, allo scopo di evidenziare le parole che assumono uno specifico significato nell'ambito del progetto. Per comprenderle in maniera corretta, si rimanda il lettore al documento “Glossario”, che contiene un elenco completo di tutte le terminologie utilizzate con relative definizioni, allo scopo di costruire un linguaggio uniforme che possa migliorare la comunicazione tra i componenti interni al gruppo e gli stakeholders^G esterni.

1 Introduzione

1.1 Scopo del documento

Questo documento definisce come le attività di controllo della qualità saranno gestite durante il ciclo di vita del prodotto software. Include dettagli sui processi e standard da seguire per assicurare che il prodotto finale rispetti i requisiti di qualità specificati. Questo piano è fondamentale per garantire che il software sviluppato sia di alta qualità e soddisfi le aspettative dei clienti e degli stakeholder. Al suo interno contiene le metriche per misurare il livello di qualità di progetto in fase di sviluppo, in modo da poter migliorare alcune procedure se giudicate non conformi alle aspettative. Si prevede che il documento abbia natura incrementale, perché le metriche potrebbero essere aggiornate o riadattate in corso di progetto, a seconda delle esigenze e delle richieste da parte del committente.

Nel corso di progetto, l'accertamento della qualità sarà documentato allegando al documento le misurazioni delle attività di verifica con le metriche descritte.

1.2 Il progetto

Il progetto nasce nell'ambito dei **sistemi gestionali di magazzino**, meglio noti con il termine inglese di *Warehouse Management Systems* (WMS), con l'obiettivo di risolvere una serie di problematiche derivanti dalle soluzioni tradizionali tuttora presenti sul mercato.

Il focus principale sarà migliorare la user experience, tramite la realizzazione di un applicativo che proponga all'utente un'interazione con il magazzino in un ambiente di lavoro 3D: questa soluzione, rispetto ai tradizionali sistemi 2D, garantirebbe una maggiore comprensione degli spazi, proponendo una visualizzazione più intuitiva e familiare del magazzino all'utente che, di conseguenza, sarà in grado di prendere decisioni organizzative più informate ed efficienti, ottimizzando i processi di logistica.

Per raggiungere questo obiettivo, l'ambiente di lavoro non può essere una semplice visualizzazione del magazzino. L'utente dovrà infatti poter:

- Navigare l'ambiente 3D;
- Progettare la scaffalatura e modificarla nel tempo;
- Inserire, spostare e rimuovere prodotti negli scaffali.

Il progetto deve concretizzarsi nella realizzazione di una web app fruibile agli impiegati d'ufficio ed incentrata sulla visualizzazione 3D del magazzino.

Per visionare il capitolato^G e la documentazione del gruppo, si veda la sezione [Riferimenti Esterni](#) del documento.

2 Metriche

2.1 Codice

Le metriche descritte nel documento saranno identificate univocamente tramite un codice standardizzato dal gruppo nella seguente maniera:

M[Tipo][Identificativo]-[Acronimo]

dove:

- **Tipo:** indica se la metrica misura la qualità di:
 - **PC:** un processo;
 - **PD:** un prodotto.
- **Identificativo:** si tratta di un numero progressivo univoco all'interno di uno stesso *Tipo*.
- **Acronimo:** indica l'acronimo del nome della metrica.

2.2 Contenuto

Per ogni metrica verranno riportati:

- Una breve **descrizione**;
- I **valori accettabili** in fase di controllo della qualità;
- I **valori ottimali** che la metrica dovrebbe raggiungere.

3 Qualità di processo

Questa sezione è dedicata alle metriche atte a misurare la qualità dei processi nel corso del progetto didattico, qui descritti seguendo lo standard **ISO/IEC 12207**, reperibile nella sezione [Riferimenti Esterni](#) del documento.

3.1 Processi primari

3.1.1 Fornitura

Uno dei compiti principali del processo di fornitura è la gestione delle procedure e delle risorse necessarie a garantire il raggiungimento degli obiettivi di progetto. Per questo motivo, le metriche di qualità adottate in questo processo hanno lo scopo di misurare come il progetto si sta muovendo rispetto alla sua pianificazione originaria, in termini di costi previsti in partenza rispetto ai costi effettivi nel corso di progetto.

Risulta subito evidente il fatto che delle semplici metriche di monitoraggio dei tempi non sono abbastanza per comprendere lo stato effettivo del progetto (molte ore di lavoro non significano sempre che il prodotto da realizzare è a buon punto) e per questo non possono garantire un buon livello di qualità. Per questo motivo, il gruppo ha scelto di utilizzare le metriche proposte dal metodo **Earned Value**, che si propone di misurare la quantità di lavoro effettivamente eseguito su un progetto.

Gli acronimi utilizzati nelle metriche faranno riferimento a quelli proposti dallo standard, reperibili al link sulle metriche di progetto nella sezione [Riferimenti Esterni](#). Per una lettura più immediata del documento si riportano i termini principali:

Acronimo	Nome	Significato
PV	Planned Value	Costo pianificato per realizzare le attività di progetto alla data corrente.
AC	Actual Cost	Costo effettivamente sostenuto alla data corrente.
EV	Earned Value	Valore delle attività realizzate alla data corrente.
BAC	Budget at Completion	Indica il valore iniziale previsto per la realizzazione del progetto.
ETC	Estimate to Complete	Valore stimato per la realizzazione delle rimanenti attività necessarie al completamento del progetto.
EAC	Estimated at Completion	Indica la revisione del BAC rispetto allo stato corrente del progetto ($EAC = AC + ETC$).
CV	Cost Variance	Calcola il valore del costo realmente maturato rispetto al costo effettivo ($CV = EV - AC$).
SV	Schedule Variance	Calcola le tempistiche effettive (se si è in anticipo o ritardo) rispetto alla schedulazione delle attività pianificate ($SV = EV - PV$).
BV	Budget Variance	Indica se alla data corrente si è speso di più o di meno rispetto a quanto previsto ($BV = PV - AC$).

Le metriche sono stabilite di conseguenza:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPC1-EAC	Si vuole che il costo attuale sia quanto più possibile in linea con quello pianificato originariamente	Errore massimo del $\pm 5\%$ rispetto a BAC	= BAC
MPC2-CV	Si vuole che il valore delle attività completate (EV) sia corrispondente o maggiore del costo sostenuto (AC)	$\geq -10\%$	≥ 0
MPC3-SV	Si vuole che il progetto produca con maggiore o uguale velocità rispetto a quanto pianificato	$\geq -10\%$	≥ 0
MPC4-BV	Si vuole che i costi previsti nella pianificazione iniziale corrispondano a quelli effettivi	Errore massimo del $\pm 10\%$	0
MPC5-PV	I costi pianificati per le attività di progetto non devono sfiorare i costi originariamente pianificati	-	La somma dei costi di volta in volta deve essere \leq BAC
MPC6-AC	I costi effettivi per le attività di progetto non devono sfiorare i costi pianificati nella revisione del BAC	-	La somma dei costi di volta in volta deve essere \leq EAC
MPC7-EV	I valore effettivo per le attività di progetto non deve sfiorare i costi pianificati nella revisione del BAC	-	La somma dei costi di volta in volta deve essere \leq EAC

3.1.2 Sviluppo

Il processo di sviluppo definisce i compiti e le attività che il gruppo deve svolgere per la realizzazione del prodotto software concorde con le esigenze del proponente ed, in particolare, si occupa delle attività di analisi dei requisiti, progettazione e codifica. Le metriche di qualità individuate per questo processo sono relative soprattutto alle ultime due attività.

3.1.2.1 Progettazione

Per assicurare qualità nella fase di progettazione si è ritenuto importante utilizzare le metriche di **Structural fan-in** e **fan-out**, applicate a procedure e file con i seguenti acronimi:

Acronimo	Nome	Significato
SFIN_p	Structural fan-in (procedure)	Numero di procedure che chiamano una specifica procedura.
SFOU_p	Structural fan-out (procedure)	Numero di procedure che una specifica procedura chiama.
SFIN_f	Structural fan-in (file)	Numero di file che hanno bisogno di uno specifico file per funzionare.
SFOU_f	Structural fan-out (file)	Numero di file da cui dipende (di cui ha bisogno) uno specifico file.

Le regole da applicare alle metriche sono dunque le seguenti:

Codice	Descrizione	Valore desiderabile
MPC8-SFINp	Si vuole che le procedure progettate vengano riutilizzate	Alto (riutilizzo di codice)
MPC9-SFOUtp	Si vuole che le procedure non siano fortemente accoppiate fra loro	Basso (bassa dipendenza tra procedure)
MPC10-SFINf	Il numero di file che utilizzano uno specifico file non deve essere troppo alto per evitare dipendenze tra file	Ragionevolmente basso (bassa dipendenza tra file)
MPC11-SFOUtf	Il livello di accoppiamento tra file deve essere basso	Basso (bassa dipendenza tra file)

3.1.2.2 Codifica

Con l'attività di codifica, i programmatori si impegnano a concretizzare quanto prodotto con l'attività di progettazione attraverso la programmazione del software vero e proprio. Lo scopo è quello di ottenere un prodotto software che rispetti i requisiti e le richieste concordati con il proponente e che ne garantisca la qualità. Per raggiungere questi obiettivi si utilizzano metriche che cercano di limitare gli errori introdotti nel codice, in particolare:

Acronimo	Nome	Significato
CCH	Code churn	Indica il numero di modifiche, aggiunte e cancellazioni apportate nel tempo ad una certa area di codice, in particolare nell'ambito di progetto, si intende modifiche apportate ad una procedura. Più sono le modifiche, più aumenta il rischio di introdurre errori nel codice. Un alto valore di nuove linee di codice potrebbe anche indicare basso riutilizzo di codice.
NB	Number of bugs	Indica il numero di bug ed errori presenti nel software.
CC	Cyclomatic complexity	Misura la complessità del codice calcolando il numero di percorsi indipendenti (contando cioè il numero di "decisioni prese" nel codice sorgente). Un alto valore di complessità ciclomantica ^G indica codice difficile da comprendere, mantenere e testare.

Le regole di qualità sono applicate alle metriche nel seguente modo:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPC12-CCH	Il numero di modifiche apportate alla singola procedura deve essere quanto più basso possibile	massimo 20% di modifiche sul codice totale della procedura	0
MPC13-NB	Si vuole che il numero di bug ed errori nel codice sia quanto più basso possibile	≤ 5	0
MPC14-CC	Il codice deve essere mantenuto semplice, facilitando comprensione, manutenzione e test	≤ 15	≤ 10

3.2 Processi di supporto

3.2.1 Documentazione

La documentazione prodotta nel corso del progetto deve essere un valido supporto al gruppo e agli stakeholders per comprendere meglio il codice prodotto, le decisioni prese e la gestione organizzativa. Per questo motivo, le metriche di qualità sulla documentazione si concentrano sulla leggibilità e sul controllo degli errori ortografici. Le metriche in questione sono le seguenti:

Acronimo	Nome	Significato
EOD	Errori ortografici per documento	Indica il numero di errori ortografici individuati per documento.
IG	Indice Gulpease	Calcola la leggibilità di un testo in lingua italiana prendendo in considerazione la lunghezza della parola e la lunghezza della frase rispetto al numero di lettere. $89 + \frac{300 * (num. \text{ frasi}) - 10 * (num. \text{ lettere})}{(num. \text{ parole})}$

Le regole di qualità sono applicate alle metriche nel seguente modo:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPC15-EOD	Si vogliono meno errori ortografici possibili per ciascun documento	≤ 5	0
MPC16-IG	La documentazione deve essere di facile lettura	30-100	40-100

3.2.2 Accertamento della qualità

Lo scopo dell'accertamento della qualità è garantire che i processi e il prodotto siano conformi alle attese e soddisfino al meglio le richieste del proponente. A tal fine e per avere una valutazione oggettiva e quantificabile della qualità, si fa riferimento al presente "Piano di Qualifica", che riporta le metriche di qualità adottate dal gruppo nell'ambito del progetto.

Diventa quindi essenziale il soddisfacimento delle metriche preposte:

Acronimo	Nome	Significato
MS	Metriche soddisfatte	Indica la percentuale di metriche di qualità soddisfatte rispetto al totale delle metriche preposte.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPC17-MS	Il numero di metriche di qualità soddisfatte deve essere molto alto	90%	100%

3.2.3 Verifica

Il processo di verifica ha lo scopo di determinare se i prodotti dei processi durante il corso del progetto rispettano i requisiti preposti. In particolare, si vuole enfatizzare la correttezza del codice prodotto attraverso le seguenti metriche:

Acronimo	Nome	Significato
CCV	Code coverage	Misura la quantità di codice (in percentuale) coperta dai test rispetto al totale del codice sorgente.
TP	Test passati	Misura la quantità di test (in percentuale) eseguiti con successo.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPC18-CCV	Si vuole che la percentuale di codice coperta da test sia molto alta	$\geq 90\%$	100%
MPC19-TP	Si vuole che tutti i test siano eseguiti con successo	100%	100%

4 Qualità di prodotto

L'obiettivo delle metriche definite in questa sezione è controllare la qualità del prodotto realizzato nel corso del progetto. Per questa ragione, diventa importante assicurarsi che il prodotto sia conforme alle richieste del proponente e che possieda le caratteristiche proprie di un software di qualità.

4.1 Customer needs

Si definiscono le seguenti metriche:

Acronimo	Nome	Significato
ROB	Copertura dei requisiti obbligatori	Indica il valore (in percentuale) di requisiti obbligatori che il prodotto soddisfa.
RDE	Copertura dei requisiti desiderabili	Indica il valore (in percentuale) di requisiti desiderabili che il prodotto soddisfa.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPD1-ROB	I requisiti obbligatori devono essere tutti soddisfatti	100%	100%
MPD2-RDE	I requisiti desiderabili devono essere soddisfatti in buona parte	70%	100%

4.2 Efficienza

Si definiscono le seguenti metriche:

Acronimo	Nome	Significato
ART	Average response time	Misura il tempo medio che il prodotto software impiega per rispondere agli input forniti dall'utente.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPD3-ART	Il prodotto deve essere veloce nel tempo di risposta	< 3 sec.	≤ 1 sec.

4.3 Usabilità

Si definiscono le seguenti metriche:

Acronimo	Nome	Significato
PC	Profondità di click per operazione	Misura il numero di click che l'utente deve fare per raggiungere un determinato obiettivo. Un numero troppo alto provoca frustrazione e difficoltà di utilizzo.
CFO	Comprensibilità funzioni offerte	Misura la quantità (in percentuale) di funzioni offerte che l'utente riesce ad individuare ed utilizzare con facilità.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPD4-PC	L'utente deve essere facilitato nell'interazione con il prodotto software	≤ 7	≤ 5
MPD5-CFO	L'utente deve comprendere facilmente quali sono e come si comportano le funzioni offerte	$\geq 90\%$	100%

4.4 Portabilità su altre piattaforme

Su richiesta del committente, il software prodotto deve essere una Web Application. Come tale, risulta essenziale assicurarsi che il prodotto sia utilizzabile tramite tutti i browser maggiormente utilizzati nel mercato.

Le applicazioni che sono state considerate sono:

- Google Chrome,
- Mozilla Firefox,
- Microsoft Edge,
- Safari,
- Opera.

Si definiscono dunque le seguenti metriche:

Acronimo	Nome	Significato
PPT	Portabilità su piattaforme	Indica la percentuale di piattaforme (tra quelle prese in considerazione) su cui il prodotto software è utilizzabile.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPD6-PPT	Il prodotto deve essere utilizzabile su tutti i browser principali.	100%	100%

4.5 Manutenibilità

Si definiscono le seguenti metriche:

Acronimo	Nome	Significato
CD	Codice duplicato	Misura la quantità (in percentuale) di codice duplicato.

Con le seguenti regole:

Codice	Descrizione	Valore accettabile	Valore ottimale
MPD7-CD	La duplicazione del codice deve essere minimizzata	$\leq 3\%$	0%

5 Test

Questa sezione è dedicata alla descrizione dei vari tipi di test che verranno effettuati per assicurare la qualità e l'affidabilità del software.

5.1 Test di Unità

I test di unità sono di basso livello e mirano a testare le singole unità dell'applicativo software in isolamento. Il concetto di unità si fa corrispondere generalmente alle funzioni per la programmazione procedurale, oppure a metodi e classi per la programmazione ad oggetti.

Sono i più economici da implementare, possono essere scritti più velocemente rispetto agli altri e generalmente vengono eseguiti in maniera automatica (rendendo veloce anche la loro esecuzione e ripetizione). Inoltre, permettono di individuare un problema in maniera più immediata, evitando che si propaghi nel software.

5.2 Test di Integrazione

I test di integrazione si pongono ad un livello più alto dei test di unità e sono eseguiti per controllare le interazioni tra le diverse componenti del software. L'obiettivo è verificare il corretto funzionamento delle componenti quando vengono integrate fra loro ed identificare problemi nelle interfacce e nelle interazioni.

Di norma sono eseguiti dopo i test di unità per verificare che le componenti che funzionano singolarmente continuino a funzionare in maniera corretta anche quando comunicano fra loro.

5.3 Test di Sistema

Sono eseguiti dopo aver completato i test sulle singole unità e sulla loro integrazione, con l'obiettivo di verificare il comportamento complessivo del sistema. Si basano sulle funzionalità espresse nei requisiti concordati con il proponente, per assicurare che tutte le richieste siano soddisfatte.

5.4 Test di Regressione

Sono eseguiti sulle nuove versioni del prodotto software per assicurarsi che le funzionalità aggiunte non abbiano introdotto errori nelle funzionalità preesistenti e che la qualità non sia regredita. Prevedono la ripetizione di tutto il processo di testing sulla nuova versione dell'applicativo, comprendente test di unità, integrazione e sistema.

5.5 Test di Accettazione

Sono test eseguiti poco prima del rilascio del prodotto, dopo aver già testato il software attraverso test di unità, integrazione e sistema. Gli obiettivi principali sono verificare che il software rispetta i requisiti prestabiliti con il cliente ed assicurarsi che sia pronto per essere rilasciato. Per questi motivi, vengono eseguiti alla presenza del committente e su un ambiente che sia quanto più vicino possibile a quello in cui si troverà a lavorare l'utente finale.

6 Resoconto

7 Riferimenti esterni

Per ulteriori chiarimenti sugli argomenti discussi nel documento, si possono consultare i seguenti link esterni:

- Capitolato **Warehouse Management 3D**:
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C5.pdf>
- **Regolamento** del progetto didattico:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>
- Link alla **documentazione del gruppo**:
<https://avant-garde-software-engineering.github.io/documentazione.html>
- Standard **ISO/IEC 12207**, versione 1995:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- Metriche di progetto, metodo **Earned Value**:
https://it.wikipedia.org/wiki/Metriche_di_progetto