

Norme di progetto

v0.6.0



<🔗>Farmacode

farmacode.swe.unipd@gmail.com

Registro delle modifiche

Versione	Data	Scrittori	Revisori	Descrizione
0.6.0	25-11-2023	Rosson Lorenzo		Realizzata prima stesura sezione 3.2, apportate modifiche alla sezione 3.1.5
0.5.0	21-11-2023	Baggio Matteo	Carraro Alessandro	Trasferimento da LaTeX a Typst del documento
0.4.0	20-11-2023	Passarella Alessandro	Carraro Alessandro	Completamento stesura sezione 3.1
0.3.0	18-11-2023	Baggio Matteo	Carraro Alessandro	Completamento stesura sezione 1
0.2.0	15-11-2023	Baggio Matteo Passarella Alessandro	Carraro Alessandro	Stesura indice
0.1.0	12-11-2023	Bomben Filippo Rosson Lorenzo	Baggio Matteo	Stesura iniziale del documento

Indice

1) Introduzione al documento	5
1.1) Scopo del prodotto	5
1.2) Glossario	5
1.3) Miglioramenti e maturità del documento	5
1.4) Riferimenti	5
2) Processi primari	5
2.1) Acquisizione	5
2.1.1) Obbiettivo	6
2.1.2) Descrizione	6
2.1.3) Valutazione dei capitolati	6
2.1.4) Valutazione delle aziende selezionate	6
2.2) Fornitura	6
2.2.1) Gara di appalto	6
2.2.2) Aggiudicazione del capitolato	6
2.2.3) Rilascio del prodotto	6
2.2.4) Gestione dei rapporti con il cliente	6
2.3) Sviluppo	6
2.3.1) Descrizione	6
2.3.2) Analisi dei requisiti	6
2.3.3) Design architetturale	6
2.3.4) Design del software	6
2.3.5) Programmazione e verifica del software	6
2.3.6) Integrazione	6
2.4) Gestione operativa	6
2.4.1) Descrizione	6
2.4.2) Utilizzo operativo	6
2.4.3) Accettazione del cliente	6
2.5) Manutenzione	6
2.5.1) Descrizione	6
2.5.2) Correzione	6
2.5.3) Adattamento	6
2.5.4) Evoluzione	6
3) Processi di supporto	7
3.1) Documentazione	7
3.1.1) Descrizione	7
3.1.2) Strumenti	7
3.1.3) Grafiche	7
3.1.4) Norme tipografiche	7
3.1.5) Struttura	7
3.1.6) Caratterizzazione	8
3.2) Gestione della configurazione	9
3.2.1) Descrizione	9
3.2.2) Versionamento	9
3.2.3) Repository	9
3.2.3.1) Struttura	9
3.3) Accertamento della qualità	10

3.3.1) Descrizione	10
3.4) Qualifica	10
3.4.1) Verifica	10
3.4.2) Validazione	10
3.5) Revisioni congiunte con il cliente	10
3.5.1) Descrizione	10
3.6) Verifiche interne	10
3.6.1) Descrizione	10
3.7) Risoluzione dei problemi	10
3.7.1) Descrizione	10
3.7.2) Gestione dei problemi	10
3.7.3) Gestione dei cambiamenti	10
3.7.4) Utilizzabilità	10
3.7.5) Valutazione del prodotto	10
4) Processi organizzativi	10
4.1) Gestione dei processi	11
4.1.1) Obbiettivo	11
4.1.2) Descrizione	11
4.1.3) Ruoli e relativa organizzazione	11
4.1.4) Gestione degli incontri	11
4.1.5) Gestione delle comunicazioni	11
4.1.6) Gestione dell'organizzazione	11
4.2) Infrastrutture	11
4.2.1) Strumenti di supporto ai processi	11
4.3) Miglioramento	11
4.3.1.1) Manutenzione migliorativa dei processi	11
4.4) Formazione	11
4.4.1) Complementi all'auto-formazione	11

1) Introduzione al documento

Questo documento è stato creato per identificare le best practices di progetto e per stabilire una metodologia di lavoro chiara nel corso dell'attività produttiva. L'obiettivo è garantire una gestione omogenea e coesa del lavoro. Per facilitare il monitoraggio del progresso e consentire un approccio incrementale, vengono registrate le diverse versioni del documento.

1.1) Scopo del prodotto

Lo scopo del prodotto è creare un'applicazione dove sia possibile verificare i possibili interessi di un cliente nei confronti di un prodotto. Al giorno d'oggi l'ambito degli e-commerce si sta sempre più espandendo ed evolvendo. La presenza di negozi virtuali permette di accedere a molti dati legati agli acquisti, alle preferenze ed al comportamento degli utenti. Questi dati se analizzati propriamente permettono di prevedere preferenze e comportamenti futuri degli utenti, dando spazio ad operazioni di marketing mirate.

Il prodotto sarà dunque un'applicazione attraverso la quale l'amministrazione di un e-commerce sarà in grado di accedere ai risultati dell'analisi dei dati relativi all'utilizzo della suddetta attività. Il lavoro principale di questa applicazione non sarà dunque svolto dal lato dell'utente, il quale avrà solo accesso ad un'analisi dei dati e potrà garantire feedback sulla loro correttezza, ma sarà svolto da un algoritmo non visibile né accessibile all'utente. Questo algoritmo utilizzerà la tecnologia dell'intelligenza artificiale per analizzare i dati forniti dall'azienda con lo scopo di trovare e definire le correlazioni tra i vari prodotti, tra i vari utenti e tra utenti e prodotti. Queste correlazioni trovate su più livelli di profondità permetteranno di creare un altro set di dati, dal quale l'utente dell'applicazione potrà accedere ai dati che necessita, principalmente questi dati saranno gli N prodotti che potrebbero interessare ad un X utente e gli N utenti che potrebbero essere interessati ad un X prodotto.

Questa applicazione inoltre per comodità d'uso sarà sviluppata sotto la forma di una webapp che potrà essere accessibile utilizzando diversi dispositivi, sistemi e browser.

1.2) Glossario

Si fa inoltre notare la presenza di un Glossario nel quale sono riportati i termini utilizzati nei documenti. Con questo principio le best practices per la creazione del prodotto riusciranno facilmente ad essere rispettate garantendo il più possibile l'omogeneità del prodotto.

1.3) Miglioramenti e maturità del documento

Questo documento è stato creato seguendo un approccio incrementale, il che implica la sua natura adattabile e suscettibile di modifiche nel tempo. Queste modifiche saranno apportate in risposta alle esigenze concordate tra i membri del gruppo e il proponente. Pertanto, questa versione del documento non deve essere considerata come una versione definitiva o completa, ma piuttosto come un punto di partenza che sarà ulteriormente sviluppato e aggiornato per meglio rispondere alle mutevoli esigenze del progetto.

1.4) Riferimenti

2) Processi primari

2.1) Acquisizione

2.1.1) Obbiettivo

2.1.2) Descrizione

2.1.3) Valutazione dei capitolati

2.1.4) Valutazione delle aziende selezionate

2.2) Fornitura

2.2.1) Gara di appalto

2.2.2) Aggiudicazione del capitolato

2.2.3) Rilascio del prodotto

2.2.4) Gestione dei rapporti con il cliente

2.3) Sviluppo

2.3.1) Descrizione

2.3.2) Analisi dei requisiti

2.3.3) Design architetturale

2.3.4) Design del software

2.3.5) Programmazione e verifica del software

2.3.6) Integrazione

2.4) Gestione operativa

2.4.1) Descrizione

2.4.2) Utilizzo operativo

2.4.3) Accettazione del cliente

2.5) Manutenzione

2.5.1) Descrizione

2.5.2) Correzione

2.5.3) Adattamento

2.5.4) Evoluzione

3) Processi di supporto

3.1) Documentazione

3.1.1) Descrizione

La documentazione software è l'insieme di informazioni, raccolte testualmente, volte allo scopo di spiegare a quali funzionalità assolve un software, come è strutturato e implementato e come lo si utilizza. Nel contesto del team di sviluppo è necessaria per facilitare il lavoro dei componenti, tenendo traccia e documentando tutti i processi e attività presenti andando a facilitare anche la manutenzione migliorando la qualità del risultato finale.

È bene quindi che vengano definite delle regole chiare e concise utili per la stesura di un documento, da seguire durante tutto il ciclo di vita del progetto allo scopo di garantire maggiore comprensione.

3.1.2) Strumenti

- Typst: scelto per la definitiva formattazione dei documenti per via della comodità con cui effettuare il versionamento dei documenti stessi;
- Overleaf (LaTeX): utilizzato nelle prime fasi del progetto per la realizzazione dei documenti necessari, successivamente cambiato con typst;
- UML: per la creazione di diagrammi UML il team ha deciso di utilizzare StarUML.

3.1.3) Grafiche

- Template: le nostre grafiche per i documenti sono state realizzate con photoshop;
- Tabelle: le tabelle presentano una classica intestazione del contenuto, i nomi delle colonne in grassetto e nessun'altra particolarità, si è scelto di utilizzare una filosofia minimale per non appesantire i documenti.

3.1.4) Norme tipografiche

- Nome file: I nomi dei file hanno tutti una notazione omogenea tra di loro, ovvero, nomi descrittivi del contenuto, lettera iniziale è sempre maiuscola e il resto tutto minuscolo, le parole sono separate da degli underscore. La data viene scritta in formato AAAA-MM-GG;
- Stile del testo: divisione in sezioni X.X.X e in caso di ulteriori suddivisioni si utilizza un elenco puntato, la sezione X.X.1 è sempre la descrizione del contenuto di quella sezione. Si cerca sempre di rendere il tutto più semplice possibile per facilitarne la lettura e mantenere ordinato il documento;
- Glossario:

3.1.5) Struttura

I documenti ufficiali hanno una struttura precisa e comune che deve essere rigorosamente rispettata per i motivi citati nella descrizione.

- Prima pagina: sempre composta dal template esclusivo del team, il logo dell'università, l'anno accademico in cui viene svolto il progetto, il nome del documento, il nome del team con la mail e i componenti;

- Registri modifiche (changelog): composti da versionamento, data della modifica effettuata, descrizione della modifica, ruolo dei componenti che hanno effettuato la modifica e i loro nomi.

Per quanto riguarda la modalità con la quale è possibile aggiornare la versione di un documento, basterà andare ad aggiungere nella sezione changelog del rispettivo file sorgente un nuovo record. Qui sotto un esempio:

changelog: (

```
"0.5.0", "21-11-2023", p.baggio, p.carraro, "Trasferimento da LaTeX a Typst
del documento",
"0.4.0", "20-11-2023", p.passarella, p.carraro, "Completamento stesura sezione
3.1",
```

)

Una volta fatto, la compilazione automatica, attuata grazie ad una github action realizzata ad hoc, insieme alle funzionalità di scripting che fornisce typst, andrà a creare effettivamente la tabella del registro delle modifiche con all'interno tutti i dati necessari. Si noti che `p` è una variabile d'ambiente in questo caso rappresentante le persone, contenente tutti i nominativi utili e ripetuti nel corso del progetto.

- Indice: ogni documento presenta un indice nella pagina seguente al registro delle modifiche, la struttura è divisa in sezioni X.X.X con il numero della pagina in cui inizia la sezione. La divisione X.X.X presenta i macro-argomenti suddivisi nei loro vari paragrafi a loro volta suddivisi in sezioni più specifiche;
- Contenuto: esposto con la maggiore chiarezza e semplicità, rigorosamente diviso in sezioni secondo i principi di indicizzazione;
- Pié pagina: solamente il numero della pagina in questione.

3.1.6) Caratterizzazione

- Formali: Sono i documenti che andranno a formare la documentazione software del prodotto. In quanto tali sono sottoposti a versionamento e a processi di verifica e approvazione. Essi comprendono documenti interni, utili quindi ai membri del team di sviluppo, ed esterni, destinati a proponente e committente.

Complessivamente ne fanno parte:

- Interni:
 - Norme di progetto, rappresentano il “way of working”;
 - Verbali interni e esterni, a uso consultativo;
- Esterni:
 - Analisi dei Requisiti;
 - Piano di Progetto;
 - Piano di Qualifica;
 - Glossario;
 - Verbali interni e esterni, attestanti di quanto discusso.
- Informali: Sono i documenti interni non destinati alla divulgazione con esterni e fini a loro stessi. Perciò non necessitano di versionamento. Spesso sono bozze in preparazione a documenti formali, o note e appunti generiche.

3.2) Gestione della configurazione

3.2.1) Descrizione

Il concetto di “gestione della configurazione” abbraccia tutte le pratiche essenziali per gestire lo stato di un prodotto software e di tutti i suoi componenti, compresi sorgenti e documentazione. Questo insieme di norme e procedure non solo fornisce informazioni sullo stato di avanzamento del progetto, ma offre anche un resoconto dettagliato dell’evoluzione nel tempo del prodotto, garantendo nel contempo che il sistema operi secondo le attese. Un’efficace gestione della configurazione è cruciale per preservare l’integrità e le prestazioni del prodotto software durante il suo avanzamento. Inoltre, dovrebbe facilitare la risoluzione di problematiche e conflitti, assicurando una gestione fluida e efficiente del ciclo di vita del software.

3.2.2) Versionamento

Il versionamento è una procedura fondamentale per la gestione di un progetto. Oltre a tracciare i cambiamenti di ogni artefatto (documento o sorgente che sia) permette il ripristino di quest’ultimo ad una sua fase precedente. Ciò rende molto più semplice la gestione di errori. Il changelog o “registro delle modifiche”, strettamente collegato al concetto di versionamento, inoltre comunica a chiunque sia il suo lettore, il ciclo di vita dell’artefatto, le modifiche effettuate, le problematiche sorte, e infine anche la distribuzione dei lavori tra i componenti del team di sviluppo.

Ogni documento oltre a essere dotato di un changelog è identificato da un numero di versione così composto:

vX.Y.Z

dove :

- X rappresenta fasi del documento che suddividono e raccolgono i cambiamenti più significativi apportati all’artefatto anche detti “major”.
- Y rappresenta modifiche minori come ad esempio la realizzazione di una sezione o feature le quali si pensa non siano sufficienti a stabilire una nuova “fase” del documento. Verranno anche identificati con l’appellativo “minor”.
- Z rappresenta piccoli aggiustamenti (fixes) o migliorie generali.

Si noti che ogni versione rappresenta non solo un’aggiunta di tipo prettamente produttivo, ma anche al sua revisione.

3.2.3) Repository

Per la gestione della configurazione e versionamento il progetto si poggia sul uso di un repository Github. Qui sotto un link alla documentazione ufficiale:

[Github Docs.](#)

3.2.3.1) Struttura

L’attuale struttura del repository è suddivisa in 3 branch:

- main;
- approval;
- sources.

main:

E’ definibile come il branch di presentazione, nel quale sono presenti solo artefatti revisionati

e approvati dal responsabile di progetto corrente. Su esso è applicata una “branch protection rule” che non ne permette i push diretti e protegge il ramo.

approval:

Come intuibile dal nome, è il branch che rappresenta il main durante lo sviluppo e che garantisce che ciò che entra nel ramo principale sia completamente revisionato e approvato. I suoi contenuti verranno uniti a quelli del main tramite un processo di merge una volta che il responsabile di progetto lo ritenga possibile. Le pull request da qualsiasi ramo verso quello di presentazione verranno infatti reindirizzate a quest’ultimo, che ne andrà a valutare la qualità, accettando il lavoro svolto o rimandandolo al mittente con direttive sul come migliorarlo.

sources:

E’ il branch relativo alla produzione della documentazione, perciò contiene solo file di tipo .typ e non è pensato per una sua supervisione esterna. I file sorgenti verranno compilati e resi disponibili automaticamente nel ramo approval.

Nel branch main è disponibile un README.md che ne descrive la struttura di cartelle. Qui sotto un link al repository:

[Repository di progetto.](#)

3.3) Accertamento della qualità

3.3.1) Descrizione

3.4) Qualifica

3.4.1) Verifica

3.4.2) Validazione

3.5) Revisioni congiunte con il cliente

3.5.1) Descrizione

3.6) Verifiche interne

3.6.1) Descrizione

3.7) Risoluzione dei problemi

3.7.1) Descrizione

3.7.2) Gestione dei problemi

3.7.3) Gestione dei cambiamenti

3.7.4) Utilizzabilità

3.7.5) Valutazione del prodotto

4) Processi organizzativi

4.1) Gestione dei processi

4.1.1) Obbiettivo

4.1.2) Descrizione

4.1.3) Ruoli e relativa organizzazione

4.1.4) Gestione degli incontri

4.1.5) Gestione delle comunicazioni

4.1.6) Gestione dell'organizzazione

4.2) Infrastrutture

4.2.1) Strumenti di supporto ai processi

4.3) Miglioramento

4.3.1.1) Manutenzione migliorativa dei processi

4.4) Formazione

4.4.1) Complementi all'auto-formazione