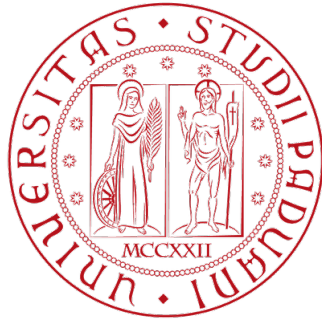


QB SOFTWARE



×



UNIVERSITÀ DEGLI STUDI DI PADOVA

CORSO DI INGEGNERIA DEL SOFTWARE

ANNO ACCADEMICO 2023/2024

Norme di progetto

CONTATTI: qbsoftware.swe@gmail.com



Registro delle modifiche

V.	Data	Membro	Ruolo	Descrizione
0.2.0	03/12/2023	A. Feltrin	Verificatore	Controllo qualità
	02/12/2023	S. Destro	Amministratore	Aggiunta sezione su processi organizzativi, sezione 3.1
0.1.0	22/11/2023	A. Giurisato	Verificatore	Controllo qualità
	22/11/2023	A. Bustreo	Amministratore	Aggiunto processo supporto per la documentazione, sezione 2.1 . Aggiunta prefazione, sezione 1



Indice

1	Prefazione	3
2	Supporting Process	4
2.1	Processo di documentazione	4
2.1.1	Implementazione del processo	4
2.1.2	Design e development	10
2.1.3	Produzione	14
2.1.4	Manutenzione	15
3	Processi organizzativi	16
3.1	Gestione organizzativa	16
3.1.1	Inizializzazione e definizione dello scopo	16
3.1.2	Pianificazione	16
3.1.3	Esecuzione e controllo	19
3.1.4	Revisione e valutazione	19
3.1.5	Chiusura	19



1 Prefazione

Con questo documento QB Software intende normare i propri processi per lo sviluppo di un progetto software. Tali processi sono stati creati a partire dallo standard ISO 12207 del 1997 [1] proposto durante il corso di ingegneria del software.

La struttura di questo documento segue lo standard [1], dove ogni processo è indicato da un numero (a)¹, ogni attività è indicata dal numero (a.b), e ogni task è indicata dalla numerazione (a.b.c). La scelta di avere un documento per struttura simile allo standard ci permette di mantenere il più fedelmente possibile le linee guida dettate dallo standard stesso. Le norme presentate in questo documento hanno lo scopo di essere il più prescrittive possibile, al fine di definire in modo "algoritmico" le procedure di lavoro, e limitare fortemente lo spazio alle scelte di libero arbitrio che rischiano di portare a situazioni meno controllate rispetto a quelle previste durante la stesura del way of working.

Prima di leggere un qualunque documento prodotto da QB Software è necessario conoscere il significato dei termini riportati nel documento: **glossario**; presente nel [repository GitHub con la documentazione di QB Software](#).

Ogni membro del gruppo si impegna a leggere, a comprendere, e a mettere in pratica in pieno le norme presenti in questo documento.

¹Eccezione per la sezione: 1 e 3.



2 Supporting Process

I processi di supporto indicati dallo standard [1] sono 8; un processo di supporto ha l'obiettivo di supportare altri processi, ad esempio quelli primari, con il fine di contribuire al successo e alla qualità di un progetto software.

2.1 Processo di documentazione

Il processo di documentazione ha lo scopo di registrare tutte le informazioni prodotte dal ciclo di vita di un processo o di un'attività. Di seguito riportiamo le quattro attività fondamentali della documentazione: implementazione del processo, design e sviluppo, produzione e manutenzione.

2.1.1 Implementazione del processo

In questa attività l'obiettivo è quello di pianificare lo sviluppo dei documenti, riportando le seguenti informazioni per ogni documento da produrre durante il progetto: il titolo del documento, il nome del documento in produzione, lo scopo, il target (a chi è rivolto il documento), le procedure che devono essere attuate, le responsabilità per l'attuazione delle procedure e le date/tempi previsti per il rilascio del documento.

Utilizzeremo il formato YYYY-MM-DD quando vogliamo indicare una data scritta nel seguente modo: anno-mese-giorno. Utilizzeremo il formato X.Y.Z per le versioni come riportato nella sezione 2.1.2. I documenti pianificati sono:

Lettera di presentazione

1. **Nome file:** Lettera_di_presentazione_X.Y.Z.pdf;
2. **scopo:** presentarsi alla candidatura di una milestone "esterna" (candidatura, RTB, PB o CA), riporta i documenti e le decisioni più rilevanti;
3. **target:** committente, documento esterno;
4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Responsabile	2.1.2
Manutenzione	Responsabile	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3



5. **schedule:** il documento deve essere redatto e approvato entro un periodo di tempo che precede di almeno tre giorni la scadenza fissata per la consegna della milestone.

Preventivo dei costi e degli impegni

1. **Nome file:** Preventivo_dei_costi_e_degli_impegni_X.Y.Z.pdf;
2. **scopo:** presentarsi alla prima milestone (candidatura), riporta i costi e la suddivisione delle ore per ruolo preventivate;
3. **target:** committente, documento esterno;
4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Responsabile	2.1.2
Manutenzione	Responsabile	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** consegna entro il 31/10/2023 alle ore 17:00.

Valutazione capitolati

1. **Nome file:** Valutazione_dei_capitolati_X.Y.Z.pdf;
2. **scopo:** presentarsi alla candidatura, valutare i 3 capitolati che risultano essere i nostri preferiti tra quelli proposti, riportando quelle che sono state le opportunità e le criticità individuate dal team durante la scelta dei capitolati;
3. **target:** committente, documento esterno;

**4. procedura e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Team QB Software	2.1.2
Manutenzione	Team QB Software	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** consegna entro il 31/10/2023 alle ore 17:00.

Norme di progetto

1. **Nome file:** Norme_di_progetto_X.Y.Z.pdf;
2. **scopo:** normare il way of working in modo prescrittivo, regolando i vari processi proposti dello standard ISO del 12207 del 1997 [1], basandosi sulle decisioni prese durante le riunioni;
3. **target:** team di QB Software, documento interno;
4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Team QB Software	2.1.2
Manutenzione	Team QB Software	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** il documento non prevede una versione finale in quanto è continuamente soggetto a modifiche e revisioni.

Glossario

1. **Nome file:** Glossario_X.Y.Z.pdf;



2. **scopo:** riportare tutti i termini di dominio utilizzate durante il progetto;
3. **target:** tutte le persone che devono interagire con questo progetto, documento esterno;

4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Team QB Software	2.1.2
Manutenzione	Team QB Software	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** il documento non prevede una versione finale in quanto è continuamente soggetto a modifiche e revisioni.

Verbalì interni

1. **Nome file:** Verbale_interno_YYYY-MM-DD_X.Y.Z.pdf;
2. **scopo:** riportare le decisioni prese durante le riunioni interne ufficiali. Il documento ha come primo obiettivo quello di riportare le decisioni di pianificazione prese, fornendo contestualmente le ragioni di tali scelte, i ticket inseriti nel ITS dovuti ai compiti assegnati e gli argomenti da trattare per la prossima riunione;
3. **target:** team di QB Software, documento interno;

4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Team QB Software	2.1.2
Manutenzione	Team QB Software	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3



5. **schedule:** il verbale deve essere redatto e verificato entro cinque giorni dall'avvenuta riunione.

Verbali esterni

1. **Nome file:** Verbale_esterno_YYYY-MM-DD_X.Y.pdf;
2. **scopo:** riportare le decisioni prese durante le riunioni esterne ufficiali. Il documento ha come primo obbiettivo quello di riportare gli argomenti di discussione durante la riunione e i ticket inseriti nel ITS dovuti alle decisioni;
3. **target:** QB Software e partecipanti esterni, documento esterno;
4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Team QB Software	2.1.2
Manutenzione	Team QB Software	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Approvazione (esterni)	Esterni	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** il verbale deve essere redatto e verificato entro cinque giorni dall'avvenuta riunione, poi l'approvazione passa a tutti i proponenti esterni.

Piano di Qualifica

1. **Nome file:** Piano_di_qualifica_X.Y.Z.pdf;
2. **scopo:** normare le procedure di verifica con l'obbiettivo di mantenere una qualità del prodotto alta;
3. **target:** QB Software, documento interno;

**4. procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Verificatore	2.1.2
Manutenzione	Verificatore	2.1.4
Verifica	Verificatore (che non ha redatto la modifica)	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** la prima versione deve essere rilasciata entro l'ultima settimana di novembre 2023, ulteriori aggiornamenti sono previsti.

Analisi dei requisiti

1. **Nome file:** Analisi_dei_requisiti_X.Y.Z.pdf;
2. **scopo:** contiene i requisiti individuati e gli use case per il prodotto da sviluppare;
3. **target:** QB Software, documento esterno;
4. **procedure e responsabilità:**

Procedura	Responsabilità	Attività di riferimento (NdP)
Sviluppo	Analisti	2.1.2
Manutenzione	Analisti	2.1.4
Verifica	Verificatore	2.1.2
Approvazione	Responsabile	2.1.3
Configuration management	Amministratore	2.1.2
Produzione e distribuzione	Responsabile	2.1.3

5. **schedule:** una prima versione del documento deve essere prodotta entro il primo incontro con il proponente, dove si deve parlare dei requisiti, la prima versione deve essere messa in produzione entro RTB.



2.1.2 Design e development

In questa attività vengono riportati tutti gli strumenti necessari allo sviluppo della documentazione, vengono definite le regole di impaginazione dei documenti da produrre. Lo scopo è quello di partire dai sorgenti `.tex` per ottenere dei documenti in formato PDF che seguano le scelte tipografiche scelte dal team QB Software.

Strumenti per lo sviluppo

- I documenti devono essere scritti in \LaTeX , usando la distribuzione [TexLive](#);
- non viene imposto nessun vincolo per l'editor/IDE da utilizzare per scrivere i documenti, comunque, si consiglia di utilizzare [Visual Studio Code](#) con l'estensione [LaTeX Workshop](#);
- il VCS da utilizzare per tracciare la storia dello sviluppo dei documenti è [Git](#);
- ogni documento deve importare il sorgente `\LaTeX base.tex`, il quale contiene tutte le utilità e le regole tipografiche normate in questo documento per lo sviluppo della documentazione;
- vengono messi a disposizione i seguenti template, presenti nel repository al path `templates/`:
 - la cartella `empty/`, struttura di un documento di base generico, da questo template derivano tutti gli altri template;
 - la cartella `verbale_interno/`, struttura di un documento per i verbali interni;
 - la cartella `verbale_esterno/`, struttura di un documento per i verbali esterni.
- [GitHub](#) come servizio per pubblicare una repository pubblica per il tracciamento della storia dei sorgenti dei documenti, nel repository `docs` ramo `develop` dove vengono caricati i documenti verificati, ma non ancora approvati;

Impaginazione di base

Ogni documento di QB Software deve essere sviluppato a partire da un template di base, presente nella cartella `src/templates/empty` nel ramo di `develop`. Il template di base deve rispettare la seguente impaginazione:

- I) deve essere in formato A4, dimensione font 12pt, margine di 2.5 cm;
- II) la prima pagina deve riportare nel seguente ordine:
 1. la scritta "QB Software";
 2. il logo di QB Software;
 3. il logo dell'università di Padova;



4. la scritta "UNIVERSITÀ DEGLI STUDI DI PADOVA";
5. la scritta "CORSO DI INGEGNERIA DEL SOFTWARE";
6. la scritta "ANNO ACCADEMICO 2023/2024";
7. il titolo del documento, e quando richiesto anche la data;
8. il contatto e-mail di QB Software come link.

III) la seconda pagina è dedicata al registro delle modifiche descritto al paragrafo [2.1.2](#);

IV) una pagina dedicata all'indice dei contenuti generato da L^AT_EX;

Ogni documento deve riportare su ogni pagina, a eccezione della prima pagina, un piè di pagina e una testatina separate dal contenuto con una linea. In ogni testatina deve essere riportato nel margine destro il logo del gruppo e nel margine sinistro la scritta "QB Software". Ogni piè di pagina deve riportare nel margine sinistro il titolo del documento e nel margine destro la pagina attuale nella seguente forma: **Pagina x di y**, dove **x** è la pagina attuale, e **y** il totale delle pagine senza contare la prima.

Regole tipografiche e di coding

Di seguito ridefiniamo, o aggiungiamo, ulteriori regole tipografiche oltre a quelle normalmente usate dal L^AT_EX, con lo scopo di rendere il documento più accessibile, ed evitare incongruenze di stile tra i diversi documenti da produrre:

- ogni tabella e figura libera presenti nel documento, a eccezione del *registro delle modifiche* e dei loghi, devono essere accompagnati da una didascalia che ne descrive il contenuto. A questo scopo è necessario usare l'ambiente LaTeX `figure` o `table` e l'istruzione `\caption` per la didascalia;
- ogni tabella e figura deve inoltre essere dotata di una label, creata con il comando `\label`. Le label devono iniziare come *fig:* per le figure, e *table:* per le tabelle;
- le tabelle vanno inserite in un ambiente `table` e devono essere posizionate sempre all'inizio della pagina, come da impostazione predefinita per l'ambiente citato, possono fare eccezione dei casi particolari dove la presenza della tabella in un certo punto del testo permette una migliore comprensione del discorso;
- quando ci si riferisce a una figura, una tabella, oppure a una sezione, citarla con il comando `\ref` specificando la tipologia (tabella, figura, sezione) dell'elemento citato seguito dal numero della sezione;
- ogni link deve essere inserito sotto forma di testo sottolineato di colore blu, inoltre non si deve scrivere direttamente l'URL, ma una frase chiara che specifichi dove quel link stia puntando;
- soltanto i link posso essere sottolineati, nessun'altra parte del testo può essere sottolineata;



- ogni sezione creata con il comando `\section` deve iniziare sempre in una nuova pagina, per fare ciò ogni section di testo va scritta in un file `.tex` a parte, sotto la cartella `sections/` e importando nel documento principale attraverso il comando `\include`;
- i riferimenti a immagini, sezioni e tabelle, interni al documento vengono colorati di azzurro, questo colore non deve essere utilizzato per nessun'altra parte del testo che non sia un link.

Registro delle modifiche

Il registro delle modifiche, per tenere una traccia completa e sensata della storia del documento deve riportare i seguenti dati:

- | | |
|--|---|
| 1. versione del documento; | 5. chi si è occupato della verifica (implicitamente il ruolo sarà sempre verificatore); |
| 2. data della modifica; | 6. data del superamento della verifica; |
| 3. membro del gruppo che ha introdotto la modifica; | 7. descrizione, breve, ma significativa delle modifiche apportante, con riferimento alla sezione modificata o introdotta. |
| 4. ruolo assunto dall'autore al momento della stesura; | |

Il registro delle modifiche deve essere implementato attraverso l'ambiente `LATEX changelog` definito all'interno del pacchetto `base.tex`. Tale ambiente deve provvedere a creare:

1. il titolo "Registro delle modifiche", il quale non verrà riportato nell'indice del documento;
2. una tabella formata dalle seguenti quattro colonne, nel seguente ordine:
 - (a) *V.*, vengono riportate le versioni del documento al momento dell'approvazione della modifica;
 - (b) *data*, vengono riportate le date di stesura della modifica e di approvazione da parte del verificatore;
 - (c) *membro*, vengono riportati gli autori della modifica, e i verificatori che hanno approvato la modifica;
 - (d) *ruolo*, vengono riportati i ruoli degli autori al momento della modifica, mentre per chi ha fatto la verifica viene riportato il ruolo di verificatore;
 - (e) *descrizione*, vengono riportate le modifiche, o aggiunte, fatte al documento facendo riferimento alle sezioni che hanno subito la modifica, o aggiunta.

Inoltre il sorgente `base.tex` fornisce il comando:

```
\newlog{Ver}{DataModifica}{Membro}{RuoloMembro}
      {DataVerifica}{Verificatore}{Descrizione}
```



che permette di inserire una nuova modifica all'interno del registro delle modifiche. Il comando deve essere usato all'interno dell'ambiente `changelog` dentro il pacchetto precedentemente citato.

Versioni documenti

La versione dei documenti proposta deriva dal [semantic versioning](#) ed è composta da 3 cifre:

$$x.y.z$$

- x indica l'approvazione e il rilascio del documento per una milestone esterna (RTB, PB o CA);
- y rappresenta una modifica sostanziale, come l'aggiunta di una nuova sezione;
- z rappresenta una modifica minore, come l'aggiornamento di un paragrafo;

Sviluppo dei documenti

Di seguito illustriamo le fasi per lo sviluppo di ogni documento. Ogni creazione/modifica di un documento deve essere collegata a una issue in modo da rendere tracciabile il lavoro svolto, vedere la sezione [2.1.2](#): automazione build per i documenti. QB Software ha individuato due momenti differenti del ciclo di vita di un documento: la creazione del documento, distinta da continue aggiunte di argomenti nuovi e sostanziali aggiunte in termini di contenuto, e la manutenzione del documento, distinta dall'assenza di modifiche sostanziali al contenuto e da modifiche che mirano a correggere o rendere il documento più fruibile.

Riportiamo la procedura per lo sviluppo di un documento:

1. il membro che deve scrivere la modifica crea un nuovo branch a partire da `develop` usando la sezione *Development* nel form della issue su GitHub;
2. i redattori sviluppano la parte richiesta seguendo le regole imposte dalle norme di progetto, ogni volta che terminano una parte del lavoro pubblicano sul proprio ramo le modifiche in modo da renderle disponibili a tutto il team di QB Software. In questa fase il documento viene considerato ancora una bozza;
3. quando un relatore ha finito la task e il documento è pronto, aggiorna il registro delle modifiche e richiede una pull request del proprio branch con il `develop`, e assegna un verificatore che dovrà verificare il lavoro;
4. il verificatore assegnato dovrà verificare i contenuti secondo quanto riportato nella sezione [2.1.2](#);
5. il verificatore se approva il contenuto deve fare il merge della pull request e elimina il branch creato dalla issue, in caso di rifiuto il redattore deve assolvere alle mancanze e iniziare nuovamente la procedura di verifica.



Verifica del documento

La verifica del documento per la fase di creazione di un documento viene fatta attraverso il metodo del *walkthrough*, cioè il verificatore legge l'interno documento e controlla:

- il pieno rispetto delle scelte tipografiche dettate dalle norme di progetto;
- la soddisfazione delle modifiche richieste dalla issue.

La verifica del documento per la fase di manutenzione avviene attraverso delle *checklist* presenti nelle Piano di Qualifica. Terminata la procedura di verifica il verificatore deve pubblicare un commento sulla pull request riportando il feedback della verifica.

Automazione build per i documenti

La configurazioni per l'automazione della build di ogni documento sono contenute dentro il file `config.csv`. Il file di configurazione appena citato deve essere gestito dall'amministratore. Oltre al file di configurazione il sistema di build usa le label assegnate alla pull request (che devono essere le stesse label assegnate alla issue) per impostare ulteriori parametri di compilazione. È responsabilità dell'autore della modifica e del verificatore nel assegnare le label corrette alla issue e alla pull request. Per la compilazione dei documenti le label da inserire sono:

1. *documentation*, indica che la pull request sta per introdurre un documento da compilare;
2. *enhancement* o *maintenance*, nel primo caso se la modifica è un aggiunta sostanziale, nel secondo caso se la modifica è di manutenzione. Permette di determinare come avviene l'aggiornamento della versione;
3. *<Sigla Documento>*, serve a identificare quale documento costruire, es: NdP (Norme di Progetto), PdP (Piano di Progetto), e così via.

Per tutte le label con la loro descrizione visitare la [repository Docs di QB Software](#).

2.1.3 Produzione

Mettere in produzione i documenti

Il responsabile può provvedere al rilascio di un documento da `develop` al `main` attraverso una pull request. L'approvazione e il conseguente rilascio di un documento deve essere fatto dal responsabile solo quando il documento è pronto per la pubblicazione per la milestone esterna. Il documento è disponibile nel branch `main` e nel [sito web di QB Software](#).

Mettere in produzione i documenti validati da esterni

I documenti che devono essere validati dagli esterni richiedono prima di seguire la procedura indicata nella sezione [2.1.3](#) il seguente requisito:

- una firma, o una qualunque marcatura che possa dimostrare l'approvazione da parte degli esterni.



Pubblicazione

I documenti approvati verranno pubblicati nel **main** della repository Docs di QB Software e nel [sito di QB Software](#).

2.1.4 Manutenzione

Ogni documento che necessita di manutenzione può avere due tipi di modifiche:

- di correzione: vengono corrette alcune parti del documento senza modificarne la struttura;
- di refactoring: viene modificata la struttura del documento, senza l'aggiunta di contenuti.



3 Processi organizzativi

3.1 Gestione organizzativa

In questa sezione vengono espone le norme relative l'organizzazione e il coordinamento delle attività interne ed esterne del gruppo, l'assegnazione dei ruoli e relativi compiti.

Come stabilito dallo standard ISO/IEC 12207:1997 [1] il processo di gestione identifica le seguenti attività:

1. Inizializzazione e definizione dello scopo [3.1.1](#);
2. pianificazione [3.1.2](#);
3. esecuzione e controllo [3.1.3](#);
4. revisione e valutazione [3.1.4](#);
5. chiusura [3.1.5](#).

3.1.1 Inizializzazione e definizione dello scopo

Il responsabile di progetto si impegna a:

- Assegnare i ruoli e i relativi compiti ai membri del gruppo;
- amministrare gli strumenti di coordinamento concordati;
- stimare i rischi;
- gestire gli imprevisti;
- gestire le comunicazioni interne ed esterne;
- calcolare i preventivi:
 1. relativo le ore e i costi divisi per ruolo;
 2. a finire, relativo il consuntivo di ogni periodo;
- determinare quando un processo, attività o task è giunto al termine e approvarlo.

3.1.2 Pianificazione

Il processo ha l'obiettivo di garantire:

- Pianificazione delle attività in modo preciso e definizione delle loro scadenze;
- monitoraggio dei progressi del team rispetto il prodotto finale;
- assegnazione dinamica ed efficace di ruoli e compiti perseguendo economicità;
- facilitazione delle comunicazioni interne ed esterne;



- stima della quantità e del tipo di risorse necessarie per ultimare il progetto nel rispetto del budget rimanente.

Ogni membro del team è tenuto a coprire almeno una volta ognuno dei ruoli elencati qui di seguito, come previsto dal *Piano dei costi*.

I ruoli predisposti per il progetto QB Software prevedono:

Responsabile

Svolge la funzione di coordinatore del gruppo e di rappresentante del team nei confronti di committente e proponente per tutta la durata del progetto. Ha molteplici responsabilità quali:

- Predisposizione e gestione delle risorse;
- coordinamento dei membri del team;
- verifica dello stato di attività e processi;
- relazioni con le figure esterne al team;
- approvazione dei documenti di progetto.

Amministratore

Definisce, controlla e amministra l'ambiente di lavoro e le risorse messe a disposizione per tutto il periodo di sviluppo del progetto. Deve accertarsi che i mezzi messi a disposizione perseguano produttività, garantendo allo stesso tempo qualità ed economicità.

Redige le *Norme di progetto* ed sua responsabilità verificare che i membri del team le seguano.

Ed è sua responsabilità:

- Amministrare infrastrutture, strumenti e documentazione;
- gestione degli imprevisti legati alla gestione dei processi;
- redigere e mantenere aggiornata la documentazione e il versionamento della stessa;
- gestire la configurazione del prodotto.

Analista

È una figura molto competente, a cui ci si affida per scomporre e approfondire le esigenze del committente. Il suo ruolo è fondamentale nelle prime fasi del progetto, in particolare per la stesura del file *Analisi dei requisiti* che sarà una serie di specifiche tecniche che saranno fondamentali per le fasi successive di sviluppo.

Si occupa di:

- Mediare fra proponenti/committenti e sviluppatori;



- studia le necessità dei proponenti definendo problemi, obiettivi e requisiti soluzione;
- etichetta i requisiti in: impliciti/espliciti e opzionali/obbligatori;
- redige i file *Studio di fattibilità* e *Analisi dei requisiti*.

Progettista

Ha l'onere di sviluppare una soluzione che soddisfi in maniera accettabile i vincoli dati dall'Analista. Effettua scelte tecniche e tecnologiche, segue lo sviluppo, non la manutenzione.

Si occupa di:

- Sviluppare un'architettura robusta seguendo le best practises perseguendo coerenza e consistenza;
- ricercare soluzioni efficienti ed efficaci che soddisfino i requisiti nel rispetto dei vincoli dati;
- decomporre il sistema in componenti e organizzarne le interazioni fra di essi;
- decomporre ruoli e responsabilità in favore di modularizzazione e riutilizzo;
- usare soluzioni ottimizzate.

Programmatore

Ha competenze tecniche e appartiene alla categoria più popolosa del gruppo. Si occupa di:

- Codificare la soluzione in modo mantenibile;
- partecipare a implementazione e manutenzione del prodotto;
- creare test ad hoc per la verifica e valutazione del codice;
- redige il manuale utente.

Verificatore

Il suo compito è quello di controllare il lavoro svolto dagli altri membri del gruppo. Inoltre ha l'onere di redigere il *Piano di qualifica*.

Deve assicurarsi che:

- I compiti vengano svolti come da programma, altrimenti offre spunti per le correzioni;
- l'esecuzione delle attività di processo non causi errori.



3.1.3 Esecuzione e controllo

Prevede che il responsabile:

- Inizi l'implementazione del piano per soddisfare gli obiettivi secondo i vincoli predisposti;
- controlli e monitori il processo fornendo report interni ed esterni come contrattato col committente;
- investighi, analizzi, riporti e risolva i problemi riscontrati durante l'esecuzione del processo. Questo può portare a cambi di programma ed è sua responsabilità assicurarsi che ogni cambiamento sia monitorato e controllato;
- riporti secondo i vincoli concordati l'avanzamento o il mancato avanzamento del processo, sia internamente che esternamente.

3.1.4 Revisione e valutazione

Prevede che il responsabile:

- Valuti prodotti e piani d'azione ai fini di soddisfare i vincoli contrattuali;
- valuti prodotti software, attività e task completate durante l'esecuzione, del processo per il raggiungimento dell'obiettivo e il completamento del piano.

3.1.5 Chiusura

Al termine dello sviluppo dei prodotti, attività e task, il responsabile determina se il processo si possa definire ultimato verificando che rispetti i vincoli contrattuali. Infine archivia documentazione e prodotti software in un ambiente predefinito da contratto.



Riferimenti bibliografici

- [1] ISO/IEC. Information technology - software life cycle processes.
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf, 1997. [Online; ultima visita 07-Novembre-2023].