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
0.4.0	17/12/2023	A. Feltrin	Verificatore	Controllo qualità
	15/12/2023	S. Rovea	Amministratore	Correzione ed espansione sezione 4.1
0.3.0	16/12/2023	S. Rovea	Verificatore	Controllo qualità
	15/12/2023	A. Feltrin	Responsabile	Aggiunta sezione su processi primari, sezione 2.1
0.2.0	03/12/2023	A. Feltrin	Verificatore	Controllo qualità
	02/12/2023	S. Destro	Amministratore	Aggiunta sezione su processi organizzativi, sezione 4.1
0.1.0	22/11/2023	A. Giurisato	Verificatore	Controllo qualità
	22/11/2023	A. Bustreo	Amministratore	Aggiunto processo supporto per la documentazione, sezione 3.1. Aggiunta prefazione, sezione 1

Indice

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.

[1] Eccezione per la sezione: 1 e 4.

2 Processi primari

I processi primari, indicati dallo standard [1], sono processi che comprendono le attività direttamente legate allo sviluppo del software.

2.1 Processo di fornitura

Il processo di fornitura comprende le attività e i compiti del fornitore. Il processo ha come obiettivo determinare le procedure e le risorse necessarie per gestire e garantire il prodotto software all'acquirente. Il fornitore gestisce il processo di fornitura seguendo il processo di gestione organizzativa e il processo di formazione.
Come stabilito dallo standard ISO/IEC 12207:1997 [1] il processo di fornitura identifica le seguenti attività:

1. avvio 2.1.1;
2. preparazione della risposta 2.1.2;
3. contrattazione;
4. pianificazione 2.1.3;
5. esecuzione e controllo 2.1.4;
6. revisione e valutazione 2.1.5;
7. consegna e completamento 2.1.6.

2.1.1 Avvio

Questa attività consiste nei seguenti compiti:

- il fornitore conduce una revisione dei vari capitolati d'appalto;
- il fornitore produce il documento Valutazione dei capitolati, che comprende per ogni capitolato valutato:
    1. una breve descrizione;
    2. il dominio applicativo;
    3. il dominio tecnologico;
    4. gli aspetti positivi;
    5. i fattori critici;
    6. conclusioni.

2.1.2 Preparazione alla risposta

Il fornitore, dopo aver scelto il capitolato, deve produrre i seguenti documenti:

- la Lettera di presentazione, che comprende:
- 1. il capitolato scelto;
  2. una lista dei documenti allegati;
  3. una lista dei membri del gruppo.
- Il Preventivo dei costi e degli impegni, che comprende:
  1. gli impegni orari per ogni ruolo;
  2. una breve considerazione per ogni ruolo;
  3. preventivo dei costi, basato sulle tariffe orarie dei ruoli;
  4. scadenza di consegna.

2.1.3 Pianificazione

Questa attività consiste nella produzione dei seguenti documenti:

- il Piano di Progetto, che comprende:
  1. analisi dei rischi: analizza i rischi che possono influenzare la pianificazione del progetto o la qualità del software;
  2. modello di sviluppo: analizza il modello scelto dal fornitore;
  3. pianificazione temporale: analizza la pianificazione temporale del progetto, e in particolare la suddivisione in fasi e la loro durata;
  4. preventivi di periodo: viene calcolata una stima del tempo di lavoro necessario e dei costi per ogni fase;
  5. consuntivi di periodo: riporta le attività effettivamente completate e i costi reali, sia in termini economici che temporali.
- Il Piano di Qualifica;
- l'Analisi dei requisiti, che comprende:
  1. introduzione: viene descritto lo scopo del documento e lo scopo del progetto. Inoltre vengono forniti i riferimenti normativi e un glossario;
  2. descrizione del prodotto;
  3. analisi dei casi d'uso obbligatori;
  4. analisi dei casi d'uso facoltativi.

2.1.4 Esecuzione e controllo

Questa attività consiste nei seguenti compiti:

- il fornitore deve implementare ed eseguire il Piano di Progetto;
- il fornitore deve sviluppare il prodotto software in conformità con il processo di sviluppo.

2.1.5 Revisione e valutazione

Questa attività consiste nell'eseguire la verifica e la validazione del prodotto software in conformità con il processo di verifica e di validazione.

2.1.6 Consegna e completamento

Questa attività consiste nei seguenti compiti:

- il fornitore deve consegnare il prodotto software;
- il fornitore deve fornire la documentazione necessaria.

3 Supporting Process

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

3.1 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.

3.1.1 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 3.1.2. 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	3.1.2
Manutenzione	Responsabile	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Responsabile	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration managment	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Team QB Software	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Team QB Software	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Team QB Software	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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

Verbali 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 obbiettivo 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	3.1.2
Manutenzione	Team QB Software	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Team QB Software	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Approvazione (esterni)	Esterni	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Verificatore	3.1.4
Verifica	Verificatore (che non ha redatto la modifica)	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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	3.1.2
Manutenzione	Analisti	3.1.4
Verifica	Verificatore	3.1.2
Approvazione	Responsabile	3.1.3
Configuration management	Amministratore	3.1.2
Produzione e distribuzione	Responsabile	3.1.3

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.

3.1.2 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 3.1.2;

IV) una pagina dedicata all'indice dei contenuti generato da LaTeX;

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

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}{Descizione}
```

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 una 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 3.1.2: 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 3.1.2;
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.

3.1.3 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 3.1.3 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.

3.1.4 Manutenzione

Ogni documento che necessità 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.

4 Processi organizzativi

I processi organizzativi indicati dallo standard [1] servono per la gestione e lo sviluppo dell'organizzazione. Le attività e i compiti in ciascuno di questi processi sono affidati all'organizzazione stessa, che ne supervisiona e garantisce l'efficacia. L'obiettivo principale è garantire un'efficace gestione complessiva delle attività.

4.1 Gestione organizzativa

In questa sezione vengono esposte 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 4.1.1;
2. pianificazione 4.1.2;
3. esecuzione e controllo 4.1.3;
4. revisione e valutazione 4.1.4;
5. chiusura **??**.

4.1.1 Inizializzazione e definizione dello scopo

Questo processo ha i seguenti obiettivi:

- pianificare con criterio le attività;
- monitorare in modo efficace il gruppo, le risorse e i processi;
- assegnare correttamente ruoli e compiti;
- facilitare la comunicazione interna ed esterna.

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 relativi alle ore e i costi divisi per ruolo;
- calcolare i consuntivi di ogni periodo;
- determinare quando un processo, attività o task è giunto al termine e approvarlo.

4.1.2 Pianificazione

Nell'attività di pianificazione il Responsabile prepara i piani dettagliati per l'esecuzione delle attività. In particolare verranno riportati:

- una descrizione completa dei compiti;
- la definizione dei tempi di completamento;
- l'assegnazione delle risorse;
- la valutazione dei rischi e relativa implementazione di misure di controllo;
- un'analisi dei costi.

Questi piani verranno redatti dal Responsabile e si troveranno all'interno del documento Piano di Progetto.

Questa sezione relativa alla pianificazione sarà strutturata come segue:

- gestione Backlog;
- gestione Sprint;
- ruoli;
- valutazione rischi;
- preventivi.

Gestione Backlog

Per la gestione delle attività è necessario rifarsi alla piattaforma Notion. Qui è presente un database contenente il Backlog completo. In particolare, sono riportati, per ogni issue:

- **titolo**: indica il nome della issue;
- **stato**: indica lo stato della issue. Ogni issue può avere soltanto uno tra i seguenti stati:
  - to-do: compito non ancora iniziato. Si noti che al momento dell'inserimento nel database, una issue ha sempre lo stato settato a to-do;
  - in-progress: compito in corso;
  - done: compito completato;
- **importanza**: indica il grado di importanza associato;
- **sprint**: indica lo Sprint in cui la issue è assegnata.

Sono inoltre presenti delle viste al database per semplificarne la visione e avere un'idea più chiara sull'andamento del progetto. Le viste presenti sono:

- **board-generica**: mostra una board generale in cui tutte le issue sono divise tra to-do, in-progress e done;
- **board-sprint**: mostra una board in cui sono presentate solo le issue relative allo Sprint in corso divise tra to-do, in-progress e done.

Gestione Sprint

All'inizio di ogni Sprint, durante la fase di pianificazione, si procede a modificare o impostare il valore del campo "sprint" all'interno del database descritto nella sezione 4.1.2. Questo si rende necessario poichè alcuni compiti vengono pianificati inizialmente, mentre altri vengono definiti successivamente.
Durante lo Sprint, è importante notare che anche in un momento successivo alla pianificazione alcune attività possono essere aggiunte nonostante non siano state originariamente pianificate. Questo avviene in base alle competenze possedute dal team e attraverso il riferimento al Backlog generale. Questa pratica permette di massimizzare il valore del lavoro svolto e mantenere una certa flessibilità.

Dopo aver selezionato quali compiti faranno parte dello Sprint, il Responsabile si occuperà di creare su GitHub una milestone relativa allo Sprint e le relative issue a cui verranno associati i vari membri del gruppo a seconda dei loro ruoli per quel determinato Sprint, descritti alla sezione 4.1.2.
Questa pratica permette di associare in modo chiaro i compiti allo Sprint in corso, consentendo al team di avere una visione chiara del lavoro e di monitorarne il progresso in maniera efficace.

Ruoli

Per garantire una suddivisione efficace delle attività, saranno delineati sei ruoli che i membri del team assumeranno durante lo svolgimento del progetto. I ruoli verranno assegnati all'inizio di ogni Sprint. Ogni membro del team è tenuto a coprire almeno una volta ognuno dei ruoli, che vengono elencati qui di seguito:

- **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:
  - 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 è sua responsabilità verificare che i membri

del team le seguano.
È inoltre 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.
  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.

Valutazione rischi
Il Responsabile si occupa di riconoscere i possibili rischi e registrarli nel documento Piano di Progetto. Una volta individuati, è fondamentale delineare una o più strategie mirate alla gestione di tali rischi.

I rischi vengono categorizzati in base alla natura del problema potenziale. Le tipologie individuate sono:

- rischi tecnologici;
- rischi legati alle persone;
- rischi organizzativi;
- rischi sulle stime;
- rischi sui requisiti.

Ogni rischio è caratterizzato da:

- intestazione: i rischi vengono identificati mediante codici univoci creati appositamente per questo scopo e un nome, in questo formato:

  `R[categoria][numero] - [nome]`

  dove:
  - `[categoria]` corrisponde alla tipologia del rischio:
    - `T` se è un rischio tecnologico;
    - `P` se è un rischio legato alle persone;
    - `O` se è un rischio organizzativo;
    - `S` se è un rischio sulle stime;
    - `R` se è un rischio sui requisiti;
  - `[numero]`: è un numero progressivo per quella categoria di rischio;
  - `[nome]`: corrisponde al nome dato al rischio.
- descrizione e conseguenze;
- valutazione della probabilità di occorrenza e della pericolosità;
- piano di controllo;
- piano di contingenza.

4.1.3 Esecuzione e controllo

L'attività di esecuzione e controllo delinea le pratiche necessarie a guidare l'attuazione del piano per raggiungere gli obiettivi, monitorare attentamente il processo e fornire report sia interni che esterni. Si definisce inoltre la gestione dei problemi emergenti, includendo l'analisi, la risoluzione e l'eventuale adattamento dei piani. Questa sezione relativa all'esecuzione e il controllo sarà strutturata come segue:

- gestione issue;
- comunicazioni interne;
- comunicazioni esterne;
- incontri interni;
- incontri esterni;
- verbali;
- controllo rischi;

Gestione issue
Nel momento in cui ad un membro del gruppo viene assegnata una issue, il procedimento è il seguente:

1. creazione del nuovo branch: il membro a cui viene assegnata la issue crea un nuovo branch a partire da `develop` usando la sezione *Development* nel form della issue su GitHub;
2. sviluppo delle attività: vengono sviluppate le attività richieste; ogni volta che una parte del lavoro è completata, le modifiche vengono pubblicate sul proprio branch, in modo da renderle disponibili a tutto il team di QB Software;
3. aggiornamento registro e pull request: quando la task è terminata, si aggiorna il registro delle modifiche e si richiede una pull request dal proprio branch verso il `develop`;
4. verifica: il verificatore si auto-assegna e dovrà verificare i contenuti secondo quanto riportato nella sezione 4.1.4;
5. approvazione o correzione delle modifiche: il verificatore, se approva il contenuto, deve fare il merge della pull request, integrando le modifiche nel branch di sviluppo principale e eliminando il branch creato per la issue. In caso di rifiuto, il membro del team a cui era assegnata la issue deve assolvere alle mancanze e iniziare nuovamente la procedura di verifica.

Comunicazioni interne
Le comunicazioni interne possono avvenire tramite:

- Whatsapp: servizio di messaggistica istantanea utilizzato dai membri del team per comunicazioni rapide e informali;
- Discord: servizio di messaggistica e videochiamate utilizzato dal team per le riunioni interne da remoto. Offre inoltre la possibilità di creare numerosi canali di comunicazione, questo risulta molto utile per strutturare la comunicazione dividendola per argomenti.

Comunicazioni esterne
Le comunicazioni esterne sono gestite dal Responsabile e possono avvenire tramite:

- posta elettronica: l'indirizzo di posta elettronica del gruppo è

  qbsoftware.swe@gmail.com

- Discord: viene creato un apposito canale di comunicazione condiviso con il proponente per questioni rapide.

Riunioni interne
Le riunioni interne del team avvengono interamente online tramite la piattaforma Discord. La data e l'ora vengono decise tramite i servizi descritti alla sezione 4.1.3. In seguito sarà compito del Responsabile creare un evento relativo all'incontro previsto su Google Calendar dell'account del gruppo per permettere a tutto il gruppo di visionarli. Le riunioni interne generano un verbale, come descritto nella sezione 4.1.3.

Riunioni esterne
Le riunioni esterne prevedono la partecipazione di almeno 5 dei membri del gruppo. Le riunioni esterne si terranno su Carbonio Chats system: piattaforma per videochiamate sviluppata dal proponente.
Le riunioni esterne generano un verbale, come descritto nella sezione 4.1.3.

Verbali
Le comunicazioni esterne e tutte le riunioni, sia interne che esterne, generano un verbale. Durante le riunioni, il Responsabile designa un membro del gruppo incaricato di prendere appunti su Notion. Questi appunti sono disponibili per tutto il gruppo e saranno utilizzati per redigere il verbale.
Il membro del gruppo incaricato della stesura del verbale viene scelto dal Responsabile.

Controllo rischi
È compito del Responsabile assicurarsi che, nel caso di attualizzazione di un rischio, il relativo piano di contingenza previsto venga seguito. È inoltre compito del Responsabile aggiornare, ogni qual volta si renda necessario, i rischi e i relativi piani

di contingenza.
Nella fase di retrospettiva di ciascuno Sprint, descritta nella sezione 4.1.4, verranno discussi i rischi incontrati.

4.1.4 Revisione e valutazione

Questa attività coinvolge la valutazione dei prodotti e dei piani d'azione per adempiere ai vincoli contrattuali, nonché la valutazione delle attività e dei prodotti software completati durante l'esecuzione del processo per il raggiungimento degli obiettivi pianificati.
Questa sezione relativa alla revisione e alla valutazione sarà strutturata come segue:

- retrospettiva;
- verifica.

Retrospettiva

Alla fine di ogni Sprint, verrà discusso con il team quanto fatto in una riunione, in particolare il Responsabile creerà un file Notion con i seguenti punti da affrontare:

- cose che hanno funzionato;
- cose che non hanno funzionato;
- rischi incontrati.

Questo documento verrà utilizzato come base di appunti per il verbale relativo alla riunione. Successivamente, il Responsabile riporterà un rapido riassunto di quanto riportato nel verbale anche nella sezione Consuntivi nel file Piano di Progetto, in particolare per i rischi incontrati durante lo Sprint e una valutazione del piano di mitigazione previsto.

Gestione oraria

Per la gestione delle ore il team utilizza un Google Sheet condiviso sull'account aziendale. Questa pratica permette di gestire in modo trasparente il tempo dedicato alle attività e facilità la valutazione della produttività.
Alla fine di ogni Sprint, il Responsabile redigerà poi un consuntivo nel documento Piano di Progetto, includendo i costi prodotti dalle ore segnate nel file in questione.

Verifica

La verifica viene fatta attraverso il metodo del *walkthrough*, cioè il verificatore legge l'intero documento o la rispettiva porzione di codice e controlla la soddisfazione delle modifiche richieste dalla issue.
La verifica per la fase di manutenzione avviene attraverso delle *checklist* presenti nel Piano di Qualifica. Terminata la procedura di verifica il verificatore deve pubblicare un commento sulla pull request riportando il feedback della verifica.

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].