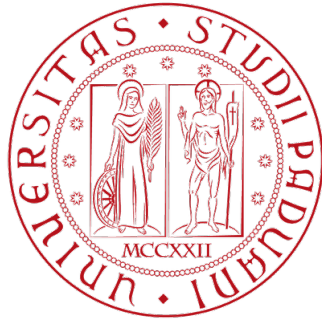


QB SOFTWARE



×



UNIVERSITÀ DEGLI STUDI DI PADOVA

CORSO DI INGEGNERIA DEL SOFTWARE

ANNO ACCADEMICO 2023/2024

---

## Norme di progetto

---

CONTATTI: [qbsoftware.swe@gmail.com](mailto:qbsoftware.swe@gmail.com)



## Registro delle modifiche

V.	Data	Autore	Ruolo	Modifiche
----	------	--------	-------	-----------



# Indice

<b>1</b>	<b>Prefazione</b>	<b>3</b>
<b>2</b>	<b>Supporting Process</b>	<b>4</b>
2.1	Processo di documentazione . . . . .	4
2.1.1	Implementazione del processo . . . . .	4
2.1.1.1	Norme di progetto . . . . .	4
2.1.1.2	Verball interni . . . . .	4
2.1.1.3	Verball esterni . . . . .	5
2.1.1.4	Piano di Qualifica . . . . .	5
2.1.1.5	Lettera di presentazione (Candidatura) . . . . .	6
2.1.1.6	Preventivo costi e degli impegni (Candidatura) . . . . .	6
2.1.1.7	Valutazione capitolati (Candidatura) . . . . .	6
2.1.1.8	Analisi dei requisiti (RTB) . . . . .	6
2.1.2	Design e development . . . . .	6
2.1.2.1	Strumenti per la stesura dei documenti . . . . .	6
2.1.2.2	Impaginazione di base . . . . .	7
2.1.2.3	Regole tipografiche . . . . .	7
2.1.2.4	Registro delle modifiche . . . . .	8
2.1.2.5	Versionamento . . . . .	9
2.1.2.6	Ciclo di vita dei documenti . . . . .	9
2.1.3	Produzione . . . . .	10
2.1.3.1	Verifica del documento . . . . .	10
2.1.3.2	Mettere in produzione il documento . . . . .	10
2.1.3.3	Configuration Managment per i documenti . . . . .	10
2.1.4	Manutenzione . . . . .	10
<b>3</b>	<b>Bibliografia</b>	<b>11</b>



# 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 proposto [1].

Questo documento si struttura in modo simile allo standard [1], ogni processo è indicato da un numero (a)<sup>1</sup>, ogni attività è indicata dal numero (a.b), e ogni task è indicata dalla numerazione (a.b.c), in fine introduciamo anche una numerazione (a.b.c.d) per fornire un'ulteriore granularità con le regole e i passi da seguire in una procedura. La scelta di avere un documento per struttura allo standard ci permette di mantenere il più fedelmente possibile le linee guida dettate dallo standard ISO. Le norme presentate in questo documento hanno lo scopo di essere il più prescrittive possibile, al fine di definire nel modo più "algoritmico" possibile le procedure di lavoro, e limitare fortemente lo spazio a 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 obbligatorio leggere il 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.

---

---

<sup>1</sup>Eccezione per la sezione: 1 prefazione.



## 2 Supporting Process

I processi di supporto indicati dallo standard [1] sono 8, un processo di supporto ha lo scopo 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 durante il ciclo di vita di un processo.

#### 2.1.1 Implementazione del processo

In questa attività normiamo la pianificazione dello sviluppo dei documenti, riportiamo tutta la documentazione da produrre durante il ciclo di vita del prodotto software.

##### 2.1.1.1 Norme di progetto

1. **Scopo**, normare il way of working in modo prescrittivo, regolando i vari processi proposti dello standard [1], basandosi sulle decisioni prese durante le riunioni;
2. **target**, team di QB Software, documento interno;
3. **procedure e responsabilità**,

Procedura	responsabilità	Sezione di riferimento (NdP)
Sviluppo	Amministratore	2.1.2.6
Manutenzione	Amministratore	2.1.4
Verifica	Verificatore	2.1.3.1

4. **schedule**, il documento non prevede una versione finale in quanto è un documento incrementale, cioè viene aggiornato quando è necessario, la prima versione è prevista per il 13/11/2023.

##### 2.1.1.2 Verbalì interni

1. **Scopo**, riportare le decisioni prese durante le riunioni interne ufficiali. Il documento ha come primo obiettivo quello di riportare le decisioni di pianificazione prese, con annesse le ragioni della decisioni, e i ticket inseriti nel ITS dovuti ai compiti assegnati, e gli argomenti da trattare per la prossima riunione;
2. **target**, team di QB Software, documento interno;



### 3. **procedure e responsabilità,**

Procedura	responsabilità	Sezione di riferimento (NdP)
Sviluppo	????	2.1.2.6
Manutenzione	-	-
Verifica	Verificatore	2.1.3.1

4. **schedule**, il verbale deve essere redatto e verificato entro 2 giorni dall'avvenuta riunione.

#### 2.1.1.3 Verballi esterni

1. **Scopo**, riportare le decisioni prese durante le riunioni esterni ufficiali. Il documento come primo obiettivo quello di riportare gli argomenti di discussione durante la riunione, e i ticket inseriti nel ITS dovuti alle decisioni;
2. **target**, QB Software, documento esterno;

### 3. **Procedure e responsabilità,**

Procedura	responsabilità	Sezione di riferimento (NdP)
Sviluppo	???	2.1.2.6
Manutenzione	-	-
Verifica	Verificatore	2.1.3.1
Approvazione	Esterni	????

4. **Schedule**: il verbale deve essere redatto e verificato entro 2 giorni dall'avvenuta riunione internamente da QB Software, poi l'approvazione passa a tutti i proponenti esterni.

#### 2.1.1.4 Piano di Qualifica

1. **Scopo**, normare le procedure di verifica;
2. **target**, QB Software, documento esterno;
3. **Procedure e responsabilità,**

Procedura	responsabilità	Sezione di riferimento (NdP)
Sviluppo	???	2.1.2.6
Manutenzione	-	-
Verifica	Verificatore	2.1.3.1
Approvazione	Esterni	????



4. **Schedule:** il verbale deve essere redatto e verificato entro 2 giorni dall'avvenuta riunione internamente da QB Software, poi l'approvazione passa a tutti i proponenti esterni.

#### 2.1.1.5 Lettera di presentazione (Candidatura)

1. **Scopo:** presentarsi alla candidatura, riportando i documenti e brevemente le decisioni più rilevanti;
2. **Target:** committente;
3. **Procedura e responsabilità:** nessuna procedura è stata attuata durante questo periodo;
4. **schedule:** deadline 31/10/2023, ore 17:00.

#### 2.1.1.6 Preventivo costi e degli impegni (Candidatura)

bla bla bla

#### 2.1.1.7 Valutazione capitolati (Candidatura)

bla bla bla

#### 2.1.1.8 Analisi dei requisiti (RTB)

bla bla bla

### 2.1.2 Design e development

In questa attività vengono riportati: tutti gli strumenti necessari allo sviluppo della documentazione, e come devono essere immaginati i documenti prodotti.

#### 2.1.2.1 Strumenti per la stesura dei documenti

- I documenti devono essere scritti in  $\text{\LaTeX}$ , usando la distribuzione [TeX Live](#);
- ogni documento deve importare il pacchetto  $\text{\LaTeX}$  `qbsoftware.sty`, 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 nella cartella `docs_src/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 tracciamento storia sorgenti dei documenti nel ramo `develop`;
- GitHub tracciamento PDF dei documenti verificati e approvati nel ramo `main`.

### 2.1.2.2 Impaginazione di base

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

- I) deve essere in formato A4, dimensione font 12pt;
- 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.
- III) la seconda pagina è dedicata al registro delle modifiche descritto al paragrafo [2.1.2.4](#);
- 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 un testatina separate dalla gabbia 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.

### 2.1.2.3 Regole tipografiche

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 documenti:

- ogni tabella e figura 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 usare l'ambiente `LaTeX figure` o `table` e l'istruzione `\caption` per la didascalia;
- ogni tabella e figura, inoltre devono avere una label che viene 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;
- quando ci si riferisce ad una figura, o a una tabella, o a una sezione, citarla con il comando `\ref` specificando la tipologia (tabella, figura, sezione) dell'elemento citato seguito dal suo numero identificativo
- ogni link deve essere inserito sotto forma di testo attraverso sottolineato di colore blu, inoltre non si deve scrivere direttamente l'URL, ma una frase chiara che specifichi dove quel link stia puntando;
- solo, e 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 importante nel documento principale attraverso il comando `\include`;
- –

#### 2.1.2.4 Registro delle modifiche

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

- versione del documento;
- data della modifica;
- autore della modifica;
- ruolo assunto dall'autore al momento della stesura;
- chi si è occupato della verifica;
- data del superamento della verifica;
- descrizione, breve, ma significativa delle modifiche apportante, con riferimento alla sezione modificata.

Il registro delle modifiche deve essere implementato attraverso l'ambiente `LATEX` `changelog` definito all'interno del pacchetto `LATEX`: `qbsoftware.sty`, 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) *Autore*, 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, 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 nel pacchetto `qbsoftware.sty` viene fornito il comando:

```
\newlog{Ver}{Data}{Autore}{RuoloAutore}{Verificatore}{Desc}
```

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 citato prima.

#### 2.1.2.5 Versionamento

La versione dei documenti proposta deriva dal [semantic versioning](#) ed è composta da 2 cifre:

$$x.y$$

- $x$  rappresenta una modifica sostanziale, come un'aggiunta di una nuova sezione, [Definire meglio];
- $y$  rappresenta una modifica minore, come l'aggiornamento di un paragrafo, o la modifica dell'impaginazione [Definire meglio];

Non è stata usata una terza cifra, in quanto il gruppo considera inutile indicare [Boh, forse togliere questa frase?].

#### 2.1.2.6 Ciclo di vita dei documenti

Di seguito illustriamo delle fasi del ciclo di vita di ogni documento, ogni creazione/modifica di un documento deve essere collegato ad una issues, [MANCA REG MOD. E UN IDEA MIGLIORE DI COME FARE LE COSE]

1. su GitHub viene creato un feature branch con GitFlow, con il nome `new_<nome_documento>`;
2. l'incaricato redige il documento, ogni volta che termina il lavoro deve fare un push del lavoro svolto sul branch, così tutto il gruppo può vedere lo stato dei lavori, e contribuire se l'attività era pianificata per più persone;
3. quando i readattori considerano le loro modifiche approvate devono fare una richiesta di pull nel ramo di develop;



4. il verificatore incaricato si segna come verificatore della pull request, e procedere a verificare il documento seguendo la checklist proposta nel Piano di Qualifica, che verifica la conformità del documento con quanto indicato nelle norme di progetto, sezione 2.1 e le richieste esposte nella issues;
5. il verificatore una volta completato la checklist, approva solo la pull request quando la checklist è completamente superata, la checklist va riportata su un commento in GitHub della pull request indicando se è stata passata o no;
6. nel caso di checklist, gli incaricati della stesura procedono a rendere conforme il documento, tornando al punto 2;
7. nel caso di approvazione, viene fatto il merge in develop, e deve essere eliminato il branch di feature in quanto è stato commitato in develop.
8. il documento viene compilato a parte, in formato PDF, e viene caricato nel main secondo la disposizione indicata dal proprio CI che si trova nel file `cm.csv`, per vedere il funzionamento leggere il processo di CM ??;

Per la manutenzione del documento vedere la sezione 2.1.4

### 2.1.3 Produzione

#### 2.1.3.1 Verifica del documento

#### 2.1.3.2 Mettere in produzione il documento

#### 2.1.3.3 Configuration Managment per i documenti

#### 2.1.4 Manutenzione



## 3 Bibliografia

### Riferimenti bibliografici

- [1] ISO. Information technology – software life cycle processes. (12207:1977), 1997.