



# ARCHI7ECHS

Archi7echs - archi7echs@gmail.com

Progetto di Ingegneria del Software  
A.A. 2024/2025

---

## Norme di Progetto

---

Autore: Il team

Ultima Modifica: 26/11/2024

Tipologia Documento: Interno

Stato: Approvato



# Tabella delle revisioni

Rev.	Data	Elaborazione	Verifica
0.1.3	26-11-2024	Leonardo Lucato dat- ta ge- stio- ne del- la board e istru- zio- ni per la re- da- zio- ne/ ve- ri- fi- ca dei do- cu- men- ti	Gabriele Checchinato
Gabriele Checchinato	0.1.2	Redatta sezione Docu- mentazione	Francesco Pozzobon
Giovanni Salvò	Giovanni Salvò	04-11-2024	Redatta sezione Introduzione
Leonardo Lucato	Gabriele Checchinato	01-11-2024	24-11-2024
Redatta la suddivisione del documento	Francesco Pozzobon	Gabriele Checchinato brie- le Chec- chi- na- to	

# Indice

1) Introduzione .....	3
1.1) Finalità del documento .....	3
1.2) Glossario .....	3
1.3) Riferimenti .....	3
1.3.1) Link al capitolato C5 - 3Dataviz .....	3
1.3.2) Slide del corso IS .....	3
2) Processi di Supporto .....	3
2.1) Documentazione .....	3
2.1.1) Modelli di documento .....	3
2.1.1.1) Documento .....	3
2.1.1.2) Allegato .....	4
2.1.1.3) Carta intestata .....	4
2.1.2) Redazione dei verbali .....	4
2.1.3) Registro delle modifiche e versionamento .....	5
2.2) Verifica e Revisione della documentazione .....	5
2.2.1) Processo per la verifica della documentazione .....	5
2.2.1.1) Relatore .....	5
2.2.1.2) Verificatore .....	6
3) Management .....	7
3.1) Gestione dell'assegnazione dei ruoli .....	7
3.2) Gestione della board .....	7
3.2.1) Processo di utilizzo board .....	7

# 1) Introduzione

## 1.1) Finalità del documento

L'obiettivo del documento è quello di definire le linee guida del gruppo per garantire un lavoro, fortemente asincrono, uniforme, coerente e di qualità. Per garantire la gestione del prodotto, composto da software e documentazione, è necessario un approccio strutturato al ciclo di vita<sub>G</sub>.

Tale documento è redatto secondo lo standard ISO 12207:1995<sub>G</sub>, il quale identifica i processi di un ciclo di vita di un software, secondo una struttura modulare con relativa responsabilità su ciascun processo.

## 1.2) Glossario

All'interno del documento saranno spesso utilizzati degli acronimi o termini tecnici per semplificare la scrittura e la lettura. Per garantire che quanto scritto sia comprensibile a chiunque, è possibile usufruire del *glossario*. Tutte le parole consultabili nel glossario saranno identificate da una «G» in colore blu. Facendo click sul collegamento si aprirà una scheda del browser con il glossario

## 1.3) Riferimenti

Il documento è stato redatto con riferimento alla seguente documentazione.

### 1.3.1) Link al capitolato C5 - 3Dataviz

- Riferimento al capitolato 5 di *Sanmarco Informatica SPA - 3Dataviz*: <https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C5.pdf> - Ultimo accesso al documento 22/11/2024

### 1.3.2) Slide del corso IS

- Riferimento alle slide IS: *Processi di ciclo di vita*: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T02.pdf> - Sezione sullo standard ISO 12207:1995 - Ultimo accesso al documento 22/11/2024
- Riferimento alle slide IS: *Regolamento del progetto didattico*: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf> - Ultimo accesso al documento 22/11/2024

# 2) Processi di Supporto

## 2.1) Documentazione

Questa sezione tratta le norme per la redazione della documentazione del gruppo, in linea con l'organizzazione del team, allineando lo stile e la gestione delle revisioni.

### 2.1.1) Modelli di documento

La redazione di tutta la documentazione del gruppo avviene in Typst<sub>G</sub> utilizzando i templates messi a disposizione nell'apposita cartella «templates» della repository<sub>G</sub>

I modelli di documento sono:

- documento
- allegato
- carta intestata

#### 2.1.1.1) Documento

Questo template<sub>G</sub> viene utilizzato per la redazione di tutta la documentazione interna ed esterna. Nella prima pagina del documento devono essere indicati, oltre a titolo e sottotitolo:

- autore del documento
- tipologia del documento (Interno<sub>G</sub> o Esterno<sub>G</sub>)
- ultima modifica
- stato del documento (Bozza oppure Approvato)

L'aggiornamento di autore e tipologia del documento è a cura del redattore<sub>G</sub> del documento.

Lo stato del documento viene posto in *Bozza* dal redattore<sub>G</sub> e aggiornato dal verificatore<sub>G</sub> quando il documento raggiunge una versione che ne consente l'approvazione<sub>G</sub> e rilascio<sub>G</sub>.

L'ultima modifica viene aggiornata automaticamente ad ogni modifica della Tabella delle revisioni<sub>G</sub>, prendendo la data dell'ultima revisione<sub>G</sub> come data di ultima modifica.

L'indice si aggiorna automaticamente in base alle sezioni di Typst, per il dettaglio su come suddividere correttamente il documento in sezioni e sottosezioni si rimanda alla documentazione ufficiale di Typst.

*Per la gestione della tabella delle revisioni si fa riferimento all'apposita sezione, Sezione 2.1.3, di questo documento.*

#### **2.1.1.2) Allegato**

Questo template<sub>G</sub> viene utilizzato per la redazione degli allegati ai verbali (interni<sub>G</sub> ed esterni<sub>G</sub>). E' compito di chi redige l'allegato indicare, nell'apposita sezione nell'intestazione del documento:

- numero allegato (num progressivo riferito al verbale)
- numero di verbale (esplicitando se interno<sub>G</sub> o esterno<sub>G</sub>)
- data del verbale

Il documento di questa tipologia viene inserito nello stesso documento del verbale.

#### **2.1.1.3) Carta intestata**

Questo template<sub>G</sub> viene utilizzato per tutte le comunicazioni ufficiali in uscita verso un destinatario esterno.

E' compito di chi redige il documento indicare, nell'apposita sezione:

- destinatario del documento
- mezzo di invio del documento
- oggetto del documento

#### **2.1.2) Redazione dei verbali**

La modalità di redazione dei verbali interni<sub>G</sub> ed esterni<sub>G</sub> è la medesima.

Nella prima pagina di contenuto, ovvero la pagina nr. 3, è necessario indicare, in ordine:

- breve sezione, scritta in *italic* con motivo e modalità della convocazione
- ordine del giorno<sub>G</sub>
- dettagli dell'incontro, con riferimento a:
  - data e ore della convocazione
  - luogo (in presenza oppure online, specificando in questo caso la piattaforma<sub>G</sub>)
  - destinatari dell'incontro
- verbale, specificando:
  - presenze

Dopo le presenze si procede con il riassunto della discussione dei relativi punti dell'OdG<sub>G</sub>, da riportare in ordine. L'ultima sezione deve sempre essere «**Varie ed eventuali**» indicando, se ci sono state, discussioni di punti extra OdG ed orario di fine dell'incontro.

Il verbale deve inoltre contenere, nella relativa sezione del template<sub>G</sub>:

- una tabella con un riassunto delle decisioni prese. Ogni riga di questa deve contenere il riferimento al punto dell'OdG<sub>G</sub>, per consentire al lettore di approfondire la sezione di interesse senza dover leggere tutto il documento, l'argomento e la decisione presa.
- una tabella TODO<sub>G</sub> con riferimento alle issue<sub>G</sub> create relativamente alle decisioni prese. In quest'ultima è necessario indicare ID<sub>G</sub> della issue<sub>G</sub>, assegnatario (se presente, in caso contrario «-»), descrizione del task<sub>G</sub>.

Alla fine del documento deve essere indicato Luogo e Data, sede del gruppo, e la data della riunione, Verbalizzante<sub>G</sub> e Responsabile di Progetto<sub>G</sub> e, nel caso di verbale esterno<sub>G</sub>, firma, per approvazione, di un rappresentante dell'azienda.

### 2.1.3) Registro delle modifiche e versionamento

La tabella contenete il registro delle modifiche<sub>G</sub>, situata a pagina 2 dei verbali e della documentazione del gruppo, escluso quindi allegati e carta intestata, deve essere aggiornata, da colui che redige il documento oppure ci effettua una modifica, ogni volta che un documento viene mandato in revisione. E' necessario indicare, in ogni riga della tabella, la data, la descrizione delle modifiche effettuate, l'autore delle modifiche e attribuire un numero di versione, secondo lo schema x.y.z<sub>G</sub>, incrementando il valore z. Il revisore<sub>G</sub>, invece, oltre ad inserire il proprio nome nell'apposita cella, è tenuto a verificare che il numero di versione sia corretto. E' a cura di quest'ultimo, quindi, valutare l'eventuale incremento del valore y. L'incremento del valore x avviene, invece, nello specifico caso del progetto, quando la documentazione viene consegnata al committente nelle due fasi di revisione: RTB<sub>G</sub> e PB<sub>G</sub>.

## 2.2) Verifica e Revisione della documentazione

Il verificatore<sub>G</sub>, una volta ricevuta la richiesta di Pull Request<sub>G</sub>, attivata secondo l'apposita procedura, è tenuto alla revisione del documento sia dal punto di vista sintattico-lessicale e grammaticale che da quello del contenuto. Compito del revisore è anche quello di controllare i corretti riferimenti del Glossario<sub>G</sub>, aggiornando i riferimenti di parole mancanti. In caso di errori di battitura o sintattici può procedere direttamente alla correzione senza modificare la tabella delle revisioni. Nel caso invece in cui le modifiche da fare riguardino il contenuto del documento quest'ultimo deve essere restituito all'autore della Pull Request<sub>G</sub> con i commenti di quanto riscontrato durante la revisione. In questo caso quindi l'iter ripartirà dalla modifica, versionamento e aggiornamento della Pull Request<sub>G</sub>.

### 2.2.1) Processo per la verifica della documentazione

Questa sezione presenta tutte le istruzioni che vengono applicate, dalla creazione/modifica del file fino alla sua verifica, per garantire la qualità del documento.

#### 2.2.1.1) Relatore

1. `git pull` — per scaricare le ultime modifiche
2. `git checkout sources` — per spostarsi sul branch di lavoro
3. `git checkout -B <nome_branch>` — per creare un nuovo branch di lavoro, partendo dal branch di lavoro sources

4. Crea dei file o modifica i file esistenti
5. `git add .` o `git add --all` — per aggiungere i file modificati nell'area di staging
6. `git commit -m "messaggio"` — per creare un commit con i file aggiunti in staging
7. `git push --set-upstream origin <nome_branch>` — per caricare il nuovo branch e le modifiche sul repository remoto
8. Aprire la Pull Request<sub>G</sub>
  - La Pull Request<sub>G</sub> può essere aperta tramite un pulsante «Create Pull Request<sub>G</sub>» presente nella pagina iniziale del repository
  - La Pull Request<sub>G</sub> può essere aperta andando nella pagina «Pull Requests», impostando «nome\_branch» come branch sorgente e «sources» come branch di destinazione. Premere successivamente il pulsante «Create Pull Request<sub>G</sub>»
  - **ATTENZIONE.** Impostare il merge al branch *sources* (viene selezionato in automatico se è stato creato il nuovo branch a partire dal branch *sources*). E' molto importante fare sempre attenzione a questo punto, per non incorrere a problemi di merge.
9. Una volta creata, si assegna il verificatore<sub>G</sub> nella sezione «Reviewers» a destra della pagina della Pull Request<sub>G</sub>, le labels, la board sotto la voce project e la milestone<sub>G</sub> se presenti.
10. Collega la/le issue/issues alla Pull Request<sub>G</sub> nella sezione «Development» a destra della pagina per la modifica della stessa. Questo permette di chiudere tutte le issue associate una volta che la Pull Request<sub>G</sub> è stata approvata.
  - **ATTENZIONE.** L'impostazione delle issue va effettuata **DOPO** la creazione della Pull Request<sub>G</sub> e non prima. Questo serve per garantire che venga aggiunto il messaggio del link tra issue e Pull Request<sub>G</sub>.

Il merge verso il branch *sources* verrà effettuata dal verificatore solo dopo la modifica/verifica del documento.

### 2.2.1.2) Verificatore

Questa sezione presenta tutte le istruzioni a cui attenersi, dal momento in cui il documento è stato modificato fino alla sua verifica.

1. `git pull` — per scaricare le ultime modifiche
2. `git checkout <nome_branch>` — per spostarsi sul branch dove ci sono le modifiche da verificare
3. Controlla i documenti che sono stati modificati
  - Se ci sono errori di battitura o sintattici, corregge il documento in locale procedendo poi con i commit
1. `git add .` o `git add --all` — per aggiungere i file modificati nell'area di staging
2. `git commit -m "messaggio"` — per creare un commit con i file aggiunti in staging
3. `git push` — per caricare le modifiche sul branch
4. Decidere se approvare o meno la Pull Request<sub>G</sub>
  - Se si decide di non approvarla per mancanza di informazioni importanti, si dovrà rifiutare la Pull Request<sub>G</sub> e indicare i motivi del rifiuto
1. Premere su «Add your review» in alto a destra
2. Premere su «Review changes» e selezionare «Request changes», scrivendo i motivi del rifiuto
3. Premere su «Submit review»

4. Attendere che il relatore<sub>G</sub> apporti le modifiche richieste
  - Se si decide di approvare la Pull Request<sub>G</sub>, procedere con i seguenti passaggi per il merge
    1. Premere su «Add your review» in alto a destra
    2. Premere su «Review changes» e selezionare «Approve»
    3. Premere su «Submit review»
    4. Premere su «Merge pull request» e successivamente su «Confirm merge»
5. Una volta effettuato il merge, comparirà un bottone «Delete branch» che permette di eliminare il ramo di lavoro. Questo passaggio è fondamentale per mantenere pulita la repository e non avere branch inutili.

## 3) Management

### 3.1) Gestione dell'assegnazione dei ruoli

#### 3.2) Gestione della board

Il team utilizza la board di GitHub per la gestione delle issue e delle attività.

Essa è suddivisa in colonne, ognuna delle quali rappresenta uno stato dell'attività.

1. **To Do:** rappresenta il nostro backlog<sub>G</sub>, ovvero tutte le attività che devono essere svolte
2. **In Progress:** attività in corso di svolgimento
3. **In review:** attività completata e in attesa di verifica
4. **Done:** attività completata e verificata

##### 3.2.1) Processo di utilizzo board

1. Assegnazione di un'attività: l'attività viene assegnata a un membro del team
  - Se c'è la presenza di un «sottogruppo» di lavoro, le decisioni relative al come suddividere le attività saranno a carico del «responsabile» del sottogruppo. Si attua quindi una sorta di «divide et impera» per garantire una maggiore efficienza e una migliore gestione delle attività.
2. Inizio dell'attività: il membro del team assegnato sposta l'attività dalla colonna **To Do** a **In Progress**
3. Completamento dell'attività: il membro del team sposta la card dalla colonna **In Progress** a **In review**
4. Verifica dell'attività: il verificatore<sub>G</sub> controlla la Pull Request<sub>G</sub> associata all'attività e, se viene approvata, per la struttura data alla repository<sub>G</sub>, l'attività verrà spostata in automatico da **In review** a **Done**

Sarà compito del responsabile<sub>G</sub> del progetto<sub>G</sub> controllare che le attività siano assegnate correttamente e che la board sia aggiornata. Inoltre, assegnerà il grado di priorità, in modo da garantire che quelle più importanti siano svolte per prime.