



ARCHI7ECHS

Archi7echs - archi7echs@gmail.com

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

Norme di Progetto

Autore: Il team

Ultima Modifica: 15/12/2024

Tipologia Documento: Interno

Stato: Approvato



Tabella delle revisioni

Rev.	Data	Descrizione	Elaborazione	Verifica
0.1.5	15-12-2024	Redatta sezione Gestione dell'assegnazione ruoli	Francesco Pozzobon, Giovanni Salvò	Gioele Scandaletti , Pietro Valdagno
0.1.4	10-12-2024	Fix sezione Verifica e Revisione della documenta- zione	Giovanni Salvò	Gabriele Checchinato, Pietro Valdagno
0.1.3	26-11-2024	Redatta gestione della board e istruzioni per la redazione/verifica dei documenti	Leonardo Lucato	Gabriele Checchinato
0.1.2	25-11-2024	Redatta sottosezione Documentazione	Francesco Pozzobon	Giovanni Salvò
0.1.1	24-11-2024	Redatta sezione Introduzione	Leonardo Lucato	Gabriele Checchinato
0.1.0	24-11-2024	Redatta la suddivisione del documento	Francesco Pozzobon	Gabriele Checchinato

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	4
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) Redattore	6
2.2.1.2) Verificatore - Responsabile	6
3) Management	7
3.1) Gestione dell'assegnazione dei ruoli	7
3.1.1) Responsabile	7
3.1.2) Amministratore	8
3.1.3) Analista	8
3.1.4) Progettista	8
3.1.5) Programmatore	8
3.1.6) Verificatore	8
3.2) Gestione della board	8
3.2.1) Processo di utilizzo board	9

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_G.

Tale documento è redatto secondo lo standard ISO 12207:1995_G, 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: *Gestione di progetto*: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T04.pdf> - Ultimo accesso al documento 12/12/2024
- Riferimento alle slide IS: *Regolamento del progetto didattico*: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf> - Ultimo accesso al documento 12/12/2024

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_G utilizzando i templates messi a disposizione nell'apposita cartella «templates» della repository_G

I modelli di documento sono:

- documento
- allegato
- carta intestata

2.1.1.1) Documento

Questo template_G 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_G o Esterno_G)
- ultima modifica
- stato del documento (Bozza oppure Approvato)

L'aggiornamento di autore e tipologia del documento è a cura del redattore_G del documento.

Lo stato del documento viene posto in *Bozza* dal redattore_G e aggiornato dal verificatore_G quando il documento raggiunge una versione che ne consente l'approvazione_G e rilascio_G.

L'ultima modifica viene aggiornata automaticamente ad ogni modifica della Tabella delle revisioni_G, prendendo la data dell'ultima revisione_G 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_G viene utilizzato per la redazione degli allegati ai verbali (interni_G ed esterni_G). 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_G o esterno_G)
- data del verbale

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

2.1.1.3) Carta intestata

Questo template_G 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_G ed esterni_G 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_G
- dettagli dell'incontro, con riferimento a:
 - data e ore della convocazione
 - luogo (in presenza oppure online, specificando in questo caso la piattaforma_G)
 - destinatari dell'incontro
- verbale, specificando:

- presenze

Dopo le presenze si procede con il riassunto della discussione dei relativi punti dell'OdG_G, 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_G:

- una tabella con un riassunto delle decisioni prese. Ogni riga di questa deve contenere il riferimento al punto dell'OdG_G, 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_G con riferimento alle issue_G create relativamente alle decisioni prese. In quest'ultima è necessario indicare ID_G della issue_G, assegnatario (se presente, in caso contrario «-»), descrizione del task_G.

Alla fine del documento deve essere indicato Luogo e Data, sede del gruppo, e la data della riunione, Verbalizzante_G e Responsabile di Progetto_G e, nel caso di verbale esterno_G, firma, per approvazione, di un rappresentante dell'azienda.

2.1.3) Registro delle modifiche e versionamento

La tabella contenete il registro delle modifiche_G, 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_G, incrementando il valore z. Il revisore_G, 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_G e PB_G.

2.2) Verifica e Revisione della documentazione

Il verificatore_G, una volta ricevuta la richiesta di Pull Request_G, 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. Il compito dell'amministratore_G inoltre, include il controllo e l'aggiornamento dei riferimenti del Glossario_G, assicurandosi che non vi siano parole mancanti. In caso di errori di battitura o sintattici può procedere direttamente il verificatore_G 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_G con i commenti di quanto riscontrato durante la revisione. In questo caso quindi l'iter ripartirà dalla modifica, versionamento e aggiornamento della Pull Request_G. Il responsabile_G inoltre, dovrà svolgere lo stesso lavoro dopo la conferma del verificatore_G per garantire l'approvazione finale. Nel caso in cui il documento che richieda approvazione sia stato redatto dal responsabile_G, l'approvazione finale viene data dall'amministratore_G, che otterrà temporaneamente il ruolo di responsabile_G per questo compito.

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) Redattore

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_G
 - La Pull Request_G può essere aperta tramite un pulsante «Create Pull Request_G» presente nella pagina iniziale del repository
 - La Pull Request_G 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_G»
 - **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_G nella sezione «Reviewers» a destra della pagina della Pull Request_G senza dimenticare d'inserire anche il responsabile_G, le labels, la board sotto la voce project e la milestone_G se presenti.
10. Collega la/le issue/issues alla Pull Request_G 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_G è stata approvata.
 - **ATTENZIONE.** L'impostazione delle issue va effettuata **DOPO** la creazione della Pull Request_G e non prima. Questo serve per garantire che venga aggiunto il messaggio del link tra issue e Pull Request_G.

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

2.2.1.2) Verificatore - Responsabile

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_G

- Se si decide di non approvarla per mancanza di informazioni importanti, si dovrà rifiutare la Pull Request_G 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_G apporti le modifiche richieste
- Se si decide di approvare la Pull Request_G, 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»

Tutte le istruzioni sopra descritte sono valide anche per il responsabile_G che dovrà inoltre seguire le indicazioni riportate di seguito (quest'ultime devono essere ignorate dal verificatore_G).

1. Premere su «Merge pull request» e successivamente su «Confirm merge»
2. 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 inutilizzati.

3) Management

3.1) Gestione dell'assegnazione dei ruoli

Il team distribuisce, in accordo con i membri, i ruoli ad ogni periodo. L'obiettivo è garantire a ciascun componente del gruppo, secondo un criterio di rotazione, l'assegnazione di ogni compito durante lo svolgimento del progetto.

I criteri che vengono considerati ad ogni scelta sono i seguenti:

- disponibilità dei singoli nel periodo seguente
- ruoli precedentemente coperti
- tendenza ad alternare i ruoli tra due periodi contigui
- possibilità di lasciare ruoli non coperti se non necessari per la fase successivamente
- possibilità di assegnare uno stesso ruolo a più membri se necessario

Vengono di seguito descritti i 6 ruoli previsti per lo sviluppo del progetto.

3.1.1) Responsabile

La figura di riferimento del gruppo e che lo rappresenta all'esterno, si occupa del coordinamento e gestione delle risorse.

Nel dettaglio la figura del Responsabile si occupa di:

- Organizzare il periodo di riferimento, assegnando ruoli e creando issue_G
- Monitorare l'andamento del periodo_G in corso mediante analisi della Project board_G e raccogliendo feedback dai diretti interessati
- Organizzare e condurre le riunioni interne del team
- Illustrare, durante i SAL_G periodici con il proponente, il lavoro svolto dal gruppo
- Predisporre il diario di bordo_G
- Valutare e gestire i rischi

- Approvare modifiche alla documentazione, secondo l'apposito procedimento
- Stesura del PdP_G con previsioni e retrospettive

3.1.2) Amministratore

Figura con il compito di assicurare l'efficienza, gestione e controllo dell'ambiente IT di lavoro nonché di supporto alla figura del Responsabile.

Nel dettaglio la figura dell'Amministratore si occupa di:

- Controllare e garantire il corretto funzionamento della repository_G
- Studiare i processi interni per renderli più efficienti
- Garantire la sicurezza della repository_G
- Aggiornare il foglio ore relativamente al periodo in corso
- Scrittura e aggiornamento delle Norme di Progetto
- Sostituire il Responsabile in caso di sua temporanea assenza
- Aggiornare il glossario
- Approvare, dopo la verifica, i documenti redatti o modificati dal Responsabile

3.1.3) Analista

Figura con il compito di analisi ed illustrazione tecnica del problema. E' richiesto, da parte di tale ruolo, la perfetta conoscenza del dominio.

Nel dettaglio la figura dell'Analista si occupa di:

- Studiare il dominio e individuare gli UC_G
- Redigere l'AdR_G in tutte le sue sezioni
- Supportare le figure del Progettista e del Programmatore

3.1.4) Progettista

Figura con il compito di individuare e determinare le scelte realizzative. E' richiesto, da parte di questa figura, competenze tecniche e tecnologiche aggiornate.

3.1.5) Programmatore

Figura con il compito di seguire la fase di codifica. Ha la responsabilità della realizzazione e mantenimento del codice. Questa figura richiede competenze tecniche ma deleghe limitate.

3.1.6) Verificatore

Figura a supporto di ogni attività del progetto. Sono richieste conoscenze e competenze tecniche e la conoscenza dettagliata delle Norme di Progetto del gruppo.

Nel dettaglio la figura del Verificatore si occupa di:

- Controllare che la documentazione redatta sia corretta, senza errori ortografici, di contenuto e che rispetti le Norme di Progetto
- Mandare in approvazione i documenti al responsabile di progetto

Per le attività in capo a tale figura si rimanda al procedimento per la gestione delle modifiche della documentazione - Sezione 2.2.1

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_G, 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_G controlla la Pull Request_G associata all'attività e, se viene approvata, per la struttura data alla repository_G, l'attività verrà spostata in automatico da **In review** a **Done**

Sarà compito del responsabile_G del progetto_G 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.