

Norme Di Progetto

RAMtastic6

18 novembre 2023



email: ramtastic6@gmail.com

Informazioni sul documento

Versione: 0.3.0
Redattori: Visentin S. Basso L.
Verificatori: Davide B. Michele Z. Leonardo B. Riccardo Z.
Destinatari: T. Vardanega, R. Cardin, Imola Informatica
Uso: Interno

Registro dei Cambiamenti - Changelog

Versione	Data	Autore	Verificatore	Dettaglio
v.0.3.0	2023-11-12	Leonardo B.	Michele Z. Samuele V. Riccardo Z.	Stesura della sottosezione 3.2 della sezione relativa alla documentazione e modifiche relative al ciclo di vita di un documento (sottosezione 3.1, parte 3.1.3).
v.0.2.0	2023-11-12	Leonardo B.	Riccardo Z.	Stesura della sezione 3.1 (Documentazione) e delle sottosezioni relative ad essa
v.0.1.0	2023-10-30	Samuele V.	Filippo T.	Prima versione

Indice

1	Introduzione	4
1.1	Riferimenti	4
1.1.1	Riferimenti normativi	4
2	Processi primari	5
3	Processi di supporto	6
3.1	Documentazione	6
3.1.1	Obiettivi	6
3.1.2	Tipologie di documenti	6
3.1.3	Ciclo di vita di un documento	6
3.1.4	Template	6
3.1.5	Struttura di un documento	7
3.1.6	Strumenti	7
3.1.7	Versionamento	7
3.2	Controllo di configurazione	7
3.2.1	versionamento	7
3.2.2	Git e Github	7
3.2.3	Struttura del repository	8
3.2.4	Controllo di flusso	8
4	Processi organizzativi	9

1 Introduzione

1.1 Riferimenti

1.1.1 Riferimenti normativi

1. Capitolato d'appalto C3:
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C3.pdf>

2 Processi primari

3 Processi di supporto

3.1 Documentazione

3.1.1 Obiettivi

3.1.2 Tipologie di documenti

I documenti prodotti possono essere classificati in due classi principali: ad uso interno e ad uso esterno; la prima categoria comprende:

- Verbali interni (i quali non necessitano di versionamento)
- Norme di progetto

La seconda categoria di documenti comprende:

- Verbali esterni
- Piano di qualifica
- Piano di progetto
- Analisi dei requisiti

3.1.3 Ciclo di vita di un documento

Un documento segue le seguenti fasi di produzione:

- Stesura: uno o più redattori si occupano di redigere il contenuto del documento.
- Verifica: ad uno o più membri del gruppo, diversi da quelli che hanno redatto il documento, viene assegnato il compito di verifica del documento tramite ITS. È importante sottolineare che tutti i documenti sopracitati sono ufficiali e devono essere, quindi, preventivamente approvati da verificatori designati.
- Approvazione: durante questa fase, il responsabile di progetto può decidere se approvare l'inclusione di un particolare documento all'interno del repository. Nel caso in cui il documento non venga approvato, si ritorna alla fase di stesura. Se quest'ultima fase va a buon fine, vengono aggiunte informazioni di versionamento secondo quanto riportato nell'apposita sezione; infine viene caricato il documento all'interno del repository.

3.1.4 Template

Il gruppo ha scelto di utilizzare template LaTeX per la produzione della documentazione. Per visualizzare la struttura e utilizzare i template, è sufficiente accedere alla cartella *documentazione-interna* all'interno del repository Github.

3.1.5 Struttura di un documento

Un documento all'interno del nostro contesto segue una struttura ben definita, le sue sezioni principali includono:

- Prima pagina: contiene il nome del gruppo e informazioni in merito al documento: uso, destinatari, redattori, verificatori, versione
- Indice: elenco strutturato dei contenuti del documento
- Registro dei cambiamenti: una tabella contenente informazioni di versionamento relative al documento attuale; queste includono: la versione, la data, l'autore, il verificatore e una breve descrizione in merito alle modifiche apportate al documento.
- Intestazione: all'interno di essa vi sono il nome e l'indirizzo email del gruppo.

N.B: I verbali non contengono il registro dei cambiamenti.

3.1.6 Strumenti

Per la creazione e la gestione della struttura dei documenti è stato deciso di utilizzare Overleaf, un editor LaTeX online che permette la stesura collaborativa dei documenti.

3.1.7 Versionamento

Il versionamento scelto per tenere traccia dei documenti è una tripletta di numeri: x.y.z.

- x è un numero intero, che fino alla release sarà < 1 , e indica la versione del progetto a cui il documento fa riferimento;
- y è un numero intero positivo, e rappresenta lo stato di verifica del documento;
- z è un numero intero positivo, e rappresenta il singolo cambiamento apportato al file.

3.2 Controllo di configurazione

3.2.1 versionamento

Capire come gestire i numeri di versione.

3.2.2 Git e Github

Il gruppo RAMtastic6 ha scelto di utilizzare come strumento di versionamento *GitHub* e di utilizzare *Git* come strumento per collegarsi alla repository GitHub. Inoltre si è scelto di utilizzare gitflow come flusso di lavoro il quale verrà discusso in modo dettagliato in seguito. Link per il download dell'installer di Git: <https://git-scm.com/downloads>.

Inoltre, a questo link: <https://rogerdudler.github.io/git-guide/index.it.html> si troverà una breve guida su come utilizzare git. In sintesi si elencano i principali comandi:

- git clone *link repo*
questo comando copierà la repository di github in locale

- `git add nome file` (oppure `."` per includere tutti i file)
git add aggiunge le modifiche apportate ai files del repository, senza eseguire questo comando un file aggiunto, eliminato o modificato non verrà salvato nella repository remota tramite il comando *git push*.
- `git commit -m "messaggio"`
salva le modifiche apportate ai files in locale associando a quello stato un messaggio
- `git push origin origine`
salva le modifiche in remoto nel branch specificato
- `git pull`
permette di aggiornare la repo in locale e in caso di necessità esegue il merge

3.2.3 Struttura del repository

La struttura della repository per i documenti deve essere:

- documenti
 - CANDIDATURA
 - RTB
 - PB
- diari_di_bordo
- documenti_interni

3.2.4 Controllo di flusso

Installando git si avrà accesso anche ai comandi di git flow.

Link tutorial di gitflow:

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

Link su come gestire le feature:

<http://danielkummer.github.io/git-flow-cheatsheet/>.

4 Processi organizzativi

Viene adottato lo snake case per nominare le cartelle all'interno del repository.