

Norme Di Progetto

RAMtastic6

28 novembre 2023



email: ramtastic6@gmail.com

Informazioni sul documento

Versione: 0.4.0
Redattori: Visentin S. Basso L. Tonietto F.
Verificatori: Tonietto F. 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.0.4.0	2023-11-19	Filippo T.	Riccardo Z.	Stesura della sotto-sottosezione 3.2.4 (relativa al controllo del flusso, con approfondimento sul flusso della documentazione prodotta) e della sottosezione 4.1 (breve analisi e descrizioni dei ruoli di progetto).
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	Scopo del documento	4
1.2	Riferimenti	4
1.2.1	Riferimenti normativi	4
1.2.2	Riferimenti informativi	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
4.1	Ruoli di progetto	9
4.1.1	Responsabile di Progetto	9
4.1.2	Amministratore di Progetto	9
4.1.3	Analista	9
4.1.4	Progettista	10
4.1.5	Verificatore	10
4.1.6	Programmatore	10

1 Introduzione

1.1 Scopo del documento

1.2 Riferimenti

1.2.1 Riferimenti normativi

1. Presentazione del capitolato d'appalto C3 - Progetto Easy Meal:
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C3.pdf>

1.2.2 Riferimenti informativi

1. Lezione "*I processi di ciclo di vita del software (T2)*" del corso di Ingegneria del Software:
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T2.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 nel branch develop.

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).

Inoltre, a questo link 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

Il gruppo RAMtastic6 ha deciso di dotarsi di Gitflow come sistema di controllo del flusso di lavoro, motivato dalla sua facilità d'uso e dalle potenzialità di gestione offerte per il repository. Con una lieve modifica nei comandi per l'esecuzione dei commit, come illustrato in questa guida su Gitflow, è possibile automatizzare il processo di creazione, gestione e chiusura di una feature. Ulteriori dettagli su come gestire le feature sono disponibili a questo link.

Gestione dei Documenti Un' particolare attenzione in tal senso è rivolta alla documentazione. Al fine di mantenere nel repository solamente i PDF dei documenti prodotti, è stato deciso di adottare la piattaforma Overleaf per la stesura in LaTeX dei documenti e la successiva verifica. Ogni volta che un documento viene redatto o aggiornato, verificato e portato alla versione corretta come precedentemente indicato, può essere comodamente convertito in formato PDF tramite Overleaf. Successivamente, il documento può essere caricato nella repository utilizzando il workflow proposto da Gitflow, con il push diretto sul branch *develop*, soprattutto quando si parla di documentazione importante e la cui stesura è in itinere.

4 Processi organizzativi

4.1 Ruoli di progetto

In questa sezione viene riportata una breve descrizione dei ruoli e delle responsabilità dei membri di un gruppo dedicato allo sviluppo di un qualsiasi tipo di *project*.

4.1.1 Responsabile di Progetto

Il Responsabile di progetto è la figura professionale, punto di riferimento sia per il committente sia per il fornitore, con lo scopo di mediare tra le due parti. Assume la responsabilità delle decisioni del gruppo dopo averle approvate.

Le sue responsabilità includono:

- Approvare l'emissione della documentazione;
- Approvare l'offerta economica sottoposta al committente;
- Pianificare e coordinare le attività di progetto;
- Gestire le risorse umane;
- Studiare e gestire i rischi.

4.1.2 Amministratore di Progetto

L'Amministratore di progetto è responsabile delle procedure di controllo e amministrazione dell'ambiente di lavoro, con piena responsabilità sulla capacità operativa e sull'efficienza.

In particolare, si occupa di:

- Ricercare, studiare e mettere in opera risorse per migliorare l'ambiente di lavoro, automatizzandolo quando possibile;
- Risolvere problemi legati alla gestione dei processi;
- Salvaguardare la documentazione di progetto;
- Effettuare il controllo di versioni e configurazioni del prodotto software;
- Redigere e attuare i piani e procedure per la gestione della qualità.

4.1.3 Analista

L'Analista è una figura con maggiori competenze riguardo al dominio applicativo del problema.

Le sue responsabilità includono:

- Studiare il problema e il relativo contesto applicativo;
- Comprendere il problema e definire la complessità e i requisiti;
- Redigere il documento di "Analisi dei Requisiti".

4.1.4 Progettista

Il Progettista gestisce gli aspetti tecnologici e tecnici del progetto.

In particolare, si occupa di:

- Effettuare scelte riguardanti gli aspetti tecnici e tecnologici del progetto, favorendone l'efficiacia e l'efficienza;
- Definire un'architettura del prodotto da sviluppare che miri all'economicità e alla manutenibilità a partire dal lavoro svolto dall'Analista;
- Redigere la *Specifica Tecnica* e la parte pragmatica del Piano di Qualifica.

4.1.5 Verificatore

Il Verificatore è responsabile della sorveglianza sul lavoro svolto dagli altri componenti del gruppo, sulla base delle proprie competenze tecniche, esperienza e conoscenza delle norme.

In particolare, si occupa di:

- Esaminare i prodotti in fase di revisione, con l'ausilio delle tecniche e degli strumenti definiti nel presente documento; item Verificare la conformità dei prodotti ai requisiti funzionali e di qualità;
- Segnalare eventuali errori.

4.1.6 Programmatore

Il Programmatore è incaricato della codifica del progetto e delle componenti di supporto che verranno utilizzate per eseguire prove di verifica e di validazione del prodotto.

In particolare, si occupa di:

- Implementare la *Specifica Tecnica* redatta dal Progettista;
- Scrivere un codice pulito e facilmente mantenibile che rispetti le norme definite nel presente documento;
- Realizzare gli strumenti per la verifica e la validazione del software;
- Redigere il Manuale Utente relativo alla propria codifica.