



Norme di progetto

swellfish14@gmail.com

Informazioni

<i>Redattori</i>	[Andrea Veronese]
<i>Revisori</i>	[Davide Porporati]
<i>Responsabili</i>	[Elena Marchioro]
<i>Uso</i>	[Interno]

Descrizione

File contenente tutte le best practices applicate al progetto

Contents

1	Introduzione	4
2	Processi Primari	4
2.1	Repository GitHub	4
2.2	Contatti con il committente/proponente	4
3	Processi di supporto	5
3.1	Organizzazione delle task	5
3.2	Documentazione	5
4	Organizzazione dei File	6
4.1	Strumenti impiegati	6
4.2	Struttura di un documento	6
4.3	Struttura di un verbale	7
5	Scrum	7
5.1	Organizzazione degli Sprint	7
6	Rotazione dei ruoli	8
7	Versionamento	8
7.1	Regole per il versionamento	8
7.2	Regole per il versionamento	9
7.3	Comandi utili di Git	9
7.4	Regole per la verifica dei documenti	10

Versione	Data	Redattore	Verificatore	Descrizione
0.0.2	23/04/2023	Davide Porporati, Elena Marchioro, Francesco Naletto	Jude Vensil Barceros	Aggiunte sezioni verifica e scrum
0.0.1	19/04/2023	Andrea Veronese	Davide Porporati	Aggiunta convenzioni adottate per lo sviluppo
0.0.0	22/03/2023	Andrea Veronese	Davide Porporati	Versione preliminare con convenzioni di base

1 Introduzione

Lo scopo del documento è quello di fornire un insieme di regole, strumenti e standard di qualità necessari per creare un Way of Working condiviso e efficacemente impiegabile da tutti i membri del gruppo. Questo documento conterrà tutte le convenzioni da utilizzare per lo sviluppo del progetto e verrà aggiornato in maniera incrementale durante tutte le fasi dello sviluppo, pertanto è da considerarsi come work-in-progress.

2 Processi Primari

2.1 Repository GitHub

Data la necessità di effettuare un lavoro sostanzioso e condiviso, il team ha deciso di utilizzare una repository pubblica su GitHub per rendere agevole la collaborazione e la visione delle modifiche apportate alle varie parti del progetto. E' stata quindi creata un'organizzazione chiamata SWellFish e al suo interno sono state create delle specifiche repository:

- SWELLFish: questa repo conterrà tutto il codice necessario per il progetto
- Documentazione: questa repo contiene tutta la documentazione di interesse per il committente e il proponente.
- dev: repository che contiene i file sorgenti necessari per la documentazione

2.2 Contatti con il committente/proponente

Per semplificare la gestione dei contatti e delle eventuali richieste con l'azienda committente e con i Professori Vardenega e Cardin, è stata creata una mail, swellfish14@gmail.com, utilizzabile da tutti i componenti del gruppo. L'uso di questa casella è stato regolamentato con un insieme di norme condivise dagli sviluppatori, pertanto un messaggio nasce dalla necessità di uno o più componenti del gruppo di avere delucidazioni da parte del committente o per sottoporre una modifica in attesa di accettazione. Il canale telegram verrà usato per chiedere informazioni in modo informale con il proponente oppure per richiedere un confronto veloce. I colloqui vengono richiesti quando

le domande da porre necessitano di una discussione diretta con il proponente. Questi ultimi vengono richiesti via mail oppure via telegram, tramite l'apposito gruppo creato con il responsabile incaricato di Imola Informatica.

3 Processi di supporto

3.1 Organizzazione delle task

Per organizzare meglio il lavoro e avere una panoramica chiara dello svolgimento delle attività e dello stato di avanzamento, è stato concordato l'uso della Project Board fornita da GitHub. Le task riguardanti la documentazione e lo svolgimento del progetto vengo assegnate ai componenti dall'amministratore ed egli si occupa inoltre di chiuderle o modificarle in base alle esigenze. Per agevolare l'attività di documentazione è stata creata una board denominata "To-Do Documenti", dove in concomitanza dello sprint settimanale vengono aggiunte nuove issue riguardanti la creazione o la modifica dei documenti da affrontare nel prossimo sprint. Per gestire le attività di sviluppo verrà utilizzata un ulteriore board chiamata "SwellFish", al cui interno verranno inserite le task da completare in un dato sprint. Queste board sono raggiungibili tramite l'apposita sezione "Projects" all'interno dell'organizzazione SwellFish.

3.2 Documentazione

Ogni documento che verrà redatto attraverserà diversi processi, ognuno dei quali vedrà impegnati uno o più componenti del gruppo. Di seguito diamo una visione d'insieme delle fasi che ogni documento attraverserà:

- Pianificazione: il file viene ideato e discusso con i membri del team sulla base delle necessità che deve soddisfare
- Stesura: il testo e il contenuto del documento viene realizzato dal redattore
- Revisione: il revisore incaricato si occupa di rileggerlo e correggere eventuali errori e/o imprecisioni, controllando che siano state rispettate le convenzioni stabilite in questo documento.

- Approvazione: il responsabile designato controlla che il documento sia corretto e coerente.

4 Organizzazione dei File

Tutta la documentazione inerente al progetto sarà incusa nella repository "Documentazione" all'interno dell'organizzazione chiamata "SwellFish". I documenti di interesse per il proponente/committente si trovano in Documentazione/esterni, mentre quelli necessari al team di sviluppo si trovano in Documentazione/interni. I verbali infine sono contenuti in Documentazione/esterni/verbali.

4.1 Strumenti impiegati

Per la redazione della documentazione il team ha concordato l'uso del linguaggio Latex. Per facilitare la produzione di documenti e ridurre i tempi necessari per la redazione, sono stati ideati dei template generici da utilizzare. Tali templates sono reperibili in "*docs/tree/dev/templates*"

4.2 Struttura di un documento

Ogni documento che non sia un verbale ha una struttura ben specifica:

- Nome del gruppo
- Logo
- Tabella con le modifiche fatte e relativo numero di versionamento (se previsto)
- Tabella con ruoli dei componenti del gruppo e finalità d'uso del documento
- Indice
- Breve descrizione dello scopo del documento

4.3 Struttura di un verbale

La struttura di un verbale utilizza la struttura di base del template impiegato per la stesura di un documento, con le seguenti aggiunte

- Data
- Ora
- Durata
- Partecipanti
- Ordine del giorno
- Riassunto contenuti

Per la nomenclatura dei verbali invece segue la seguente struttura:

- verbale
- nome azienda/docente/attività svolta
- data

5 Scrum

Il team utilizzerà questo framework per gestire le varie fasi di sviluppo del progetto.

5.1 Organizzazione degli Sprint

Ogni sprint avrà le seguenti caratteristiche:

- Durata: una settimana. Inizia il venerdì e termina il venerdì successivo con la riunione concordata
- Riunione: viene effettuata il venerdì al termine dello sprint e vengono discussi i seguenti punti:
 - Resoconto delle attività completate
 - Problemi riscontrati e soluzioni adottate

- Analisi e resoconto delle ore e dei costi sostenuti
- Pianificazione data inizio prossimo sprint e obiettivi da fissare

Gli obiettivi pianificati per lo sprint successivo verranno tradotti in "item" da aggiungere alla/e board presenti su Github e ad ogni item verranno assegnati dei membri del gruppo, coerentemente con la suddivisione dei ruoli per lo sprint in questione.

6 Rotazione dei ruoli

Vista l'adozione del framework scrum, il team ha deciso di far ruotare i ruoli dei componenti in corrispondenza della fine dello sprint, seguendo l'ordine alfabetico del nome dei componenti. Pertanto ogni membro del team avrà impegni e responsabilità diverse ogni settimana. Questo approccio permette di ottenere un maggiore coinvolgimento.

7 Versionamento

Il VCS scelto dal gruppo per facilitare la gestione delle modifiche è Git, mediante l'utilizzo di GitHub. A questo scopo è stata creata un'organizzazione denominata "SwellFish", al cui interno sono state create le opportune repository in base alle necessità del progetto. Le varie repository presenti nell'organizzazione sono raggiungibili al seguente link : "[https : //github.com/orgs/SWElfish14/repositories](https://github.com/orgs/SWElfish14/repositories)".

7.1 Regole per il versionamento

Come prassi per la nomenclatura dei file il gruppo ha concordato la seguente struttura condivisa:

- nome file: tutto in minuscolo, con le parole separate dall'underscore
- numero versione

Per la verifica e la validazione dei file il gruppo ha deciso di utilizzare la seguente prassi:

- Creazione di un nuovo branch per un apposito documento
- Stesura del documento

- Caricamento sul branch appena creato
- Verifica della struttura, contenuti e rispetto delle convenzioni stabilite
- Pull request sul branch dev
- Rilascio sul branch master

7.2 Regole per il versionamento

E' stato concordato l'uso del Versionamento Semantico, affidandosi alle caratteristiche principali. La base del versionamento ha struttura "X.Y.Z" e le seguenti regole verranno usate per indicare una modifica minore o sostanziale del documento.

- Major Zero: E' la versione 0.X.Y, denota una versione preliminare del documento
- Patch Z: un aumento del valore della "Z" indica che sono state introdotte delle modifiche al testo da parte del redattore
- Minor Y: l'aumento della 'Y' indica una modifica testuale che è stata approvata dal verificatore
- Major X: l'incremento della 'X' si ha se è stata pubblicata dal responsabile una modifica non compatibile con le versioni precedenti del documento

7.3 Comandi utili di Git

I comandi più frequentemente utilizzati sono riassunti in seguito:

- Sincronizzazione con repository remota: `git pull`;
- Creazione di un nuovo branch: `git branch nome_branch`;
- Passaggio a un branch specificato: `git checkout nome_branch`;
- Aggiunta delle modifiche alla stage area: `git add file_da_aggiungere`;
- Creazione del commit con le modifiche: `git commit`;

- Push sul remote di un nuovo branch: `git push --set-upstream origin nome branch`;
- Push su un branch già esistente: `git push`

7.4 Regole per la verifica dei documenti

Il verificatore incaricato utilizzerà la seguente procedura per valutare l'idoneità di un documento:

- Controllo della pull request da parte degli altri membri del gruppo
- Viene verificato che il documento prodotto si attenga alle linee guida presenti in questo documento
- Viene verificata la qualità del contenuto
- Se i precedenti punti sono soddisfatti viene effettuato il merge nel main branch della repo Documentazione