



ERROR_418

Verbale 29/10/23

Mail: error418swe@gmail.com

Redattori: Antonio Oseliero, Alessio Banzato

Verificatori: Riccardo Carraro, Giovanni Gardin, Rosario Zaccone

Amministratori: Silvio Nardo, Mattia Todesco

Destinatari: T. Vardanega, R. Cardin

Inizio Meeting: 15:00 Fine Meeting: 16:30 Durata: 1:30h

Presenze:

Nome	Durata Presenza
Antonio	1:30h
Alessio	1:30h
Rosario	1:30h
Giovanni	1:30h

Ordine del giorno:

- panoramica dei comandi base di git;
- panoramica delle funzioni di collaborazione di GitHub;
- discussione sulle strategie di branching da adottare.

1 git e Github

Si è effettuata una panoramica sui comandi git di base come init, add, push, pull, status, branch, checkout e sulle funzioni di pull request di GitHub.

2 Strategie di branching

Si sono confrontate due strategie di branching:

- **main-feature:** un ramo mainline e feature branch atomici e indipendenti;
- **main-dev-feature:** un ramo mainline, un ramo di sviluppo e feature branch atomici e indipendenti.

Dalla discussione è emerso che la strategia 1 è da preferirsi per progetti rilasciati in modalità rolling, senza versionamento. Tuttavia, poiché il dominio d'applicazione del capitolato è on-premise, appare più adatto un approccio con versionamento (strategia 2). Inoltre, il proponente potrebbe avere la necessità di fornire versioni del software diverse. Per concludere, lo stack tecnologico che sarà verosimilmente adottato avrà numerose dipendenze verso librerie esterne, che dovranno essere congelate prima del rilascio.

Per questi motivi, si è scelto di approcciare lo sviluppo seguendo la strategia 2. Non si tratta comunque di una scelta vincolante perché sarà sempre possibile commutare la strategia.

Si sottolinea comunque l'enorme importanza di progettare feature piccole, ben delineate e atomiche.