WorkFlow Git

GitFlow:

A diagram of a diagram

Description automatically generated

- Nhánh master: là nhánh chứa phiên bản ổn định, mỗi commit trên nhánh main đều chứa 1 phiên bản của dự án.

- Nhánh develop: là nhánh chứa mã nguồn đang trong quá trình phát triển là nơi lập trình viên làm việc thường xuyên và các chức năng(Feature) sẽ được sáp nhập liên tục vào nhánh.

- Nhánh Feature: các chức năng luôn được hoạt động trên 1 nhánh riêng được gọi là Feature, sau khi hoàn thành cách Feature sẽ được sáp nhập vào develop giúp quản lý xung đột.

- Nhánh release: là nhánh được tạo ra từ nhánh develop dùng để thực hiện các bước chuẩn bị cuối cùng trước khi phát hành như kiểm tra tổng quan sửa các lỗi phát sinh hay cập nhật tài liệu, sau đó nhánh release sẽ được sáp nhập vào nhánh main để phát hành phiên bản mới bên cạnh đó nhóm realease sẽ được sáp nhập vào nhóm develop để đảm bảo mã nguồn tại môi trường phát triển và môi trường sản phẩm là giống nhau.

- Nhánh Hotfix: nếu phát hiện lỗi trên phiên bản nhánh hotfix sẽ được tạo ra từ nhánh master để sửa lỗi sau khi sửa xog nhánh hotfix sẽ được sáp nhập vào nhánh master và develop để sửa lỗi

Hoạt động: master là nhánh mặc định khi khởi tạo repository, sau đó sẽ tạo nhánh develop từ nhánh master, develop sẽ là nơi các chức năng mới sẽ được sáp nhập liên tục, các nhánh feature sẽ được tạo ra để tạo các chức năng sau khi hoàn thành các nhóm sẽ được sáp nhập lại vào main. Khi hoàn thành và muốn phát hành phiên bản trước khi phát hành nhánh release sẽ được tạo để kiểm tra và sửa lỗi trực tiếp sau đó nhánh release sẽ được hợp nhất vào nhánh master và develop, nếu có lỗi phát sinh trong quá trình vận hành cần xử lý sẽ phải tạo nhánh hotfix từ master để sửa lỗi sau khi sửa xog sáp nhập hotfix và master và develop.

=> gitflow giúp tổ chức công việc khoa học giảm xung đột và đảm bảo chất lượng sản phẩm

GIT

1. Làm thế nào để hoàn tác một commit chưa push lên remote?

- Sử dụng git reset để đưa về các commit trước và xóa các commit sau commit đó

VD: có các commit như first, second, third và hiện tại đang ở third

A black screen with white text

Description automatically generated

* sử dụng câu lệnh git reset –soft với hash của commit first sẽ đưa về

A black screen with white text

Description automatically generated

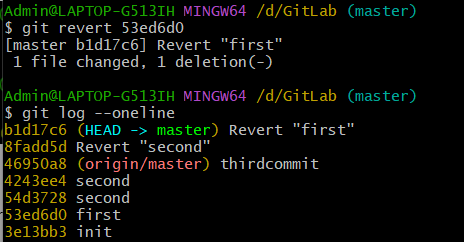
2. Làm thế nào để hoàn tác một commit đã push lên remote?

- Sử dụng git revert để đưa commit trước lên đầu mà không bị xóa lịch sử commit sẽ không ảnh hưởng đến commit trước

A screen shot of a computer

Description automatically generated

* Sử dụng lệnh $ git revert 53ed6d0 để hoàn tác về commit first



3. Làm thế nào để apply một commit từ branch này sang branch khác?

- Sử dụng git cherry-pick + hashcommit để apply

A screen shot of a computer program

Description automatically generated

A black background with colorful text

Description automatically generated

Sử dụng lệnh $ git cherry-pick edb6699 để apply commit testDev2 sang branch master

A screen shot of a computer code

Description automatically generated

4. Bạn đang phát triển chức năng trên branch feature, làm thế nào để có những thay đổi mới nhất từ branch master về branch hiện tại?

- để có thay đổi mới nhất sẽ cần phải git check out từ branch hiện tại và sử dụng merge để lấy những thay đổi mới về

5. Conflict xảy ra khi nào?

- Conflict xảy ra khi Git không thể tự động hợp nhất (merge) các thay đổi từ hai branch khác nhau, thường là do cùng một phần của file được thay đổi ở cả hai branch.

6. Nêu ít nhất một điểm giống nhau và một điểm khác nhau giữa Merge và Rebase.

- Giống nhau: Cả hai đều dùng để tích hợp các thay đổi từ một branch vào một branch khác.

- Khác nhau:

* Merge: Tạo một commit merge mới, giữ nguyên lịch sử commit của cả hai branch.
* Rebase: Di chuyển các commit của branch hiện tại lên đầu branch target, làm cho lịch sử commit thẳng hơn.

7. Fetch và Pull khác gì nhau?

- Git fetch: Lấy thay đổi từ remote về local nhưng không hợp nhất vào branch hiện tại.

- Git pull: Lấy thay đổi từ remote và tự động merge vào branch hiện tại.

8. Tôi vừa thay đổi file trong dự án và đã commit, bây giờ tôi lại phát hiện ra một lỗi nhỏ và muốn sửa file mà không phải thêm một commit nữa thì làm thế nào?

- sử dụng git add +file sửa lỗi sau đó sử dụng git commit –amend

9. Bạn đang đứng ở branch `master` làm thế nào để tạo ra một branch mới base trên branch `feature/abc`.

- sử dụng git checkout -b <new-branch> feature/abc

10. Tôi vừa thay đổi file trong dự án và đã commit nhưng chưa push lên remote, bây giờ tôi muốn hoàn tác thao tác commit và chuyển những thay đổi trong commit về lại working tree thì phải làm thế nào?

- sử dụng git reset --soft + tên commit muốn hoàn tác

Btap tình huống:

Ex1:

* Pull code từ repo:

A computer screen shot of a computer screen

Description automatically generated

* Switch sang nhánh f1\_{ten\_cua\_ban}:

A screen shot of a computer

Description automatically generated

* Commit code lên hệ thống

A computer screen shot of text

Description automatically generated

* Remote Nhánh f1\_minhhieu đã có commit

A screenshot of a computer

Description automatically generated

Ex2:

* Tạo 1 nhánh code branch ra từ master tên duy nhất theo dạng feature\_xxxx

A black background with white text

Description automatically generated

* Tạo file rocket-01.txt
* Thêm nội dung : rocket 01h vào file rocket-01.txt

A screenshot of a computer

Description automatically generated

* Commit lên hệ thống và push lên hệ thống

A computer screen shot of a black screen

Description automatically generated