

## Báo cáo Nhóm - Buổi 3

### 1. Quy trình làm việc nhóm

Nhóm chúng tôi gồm 3 thành viên: Toàn (trưởng nhóm), Thiên và Dũng.  
Ngay từ đầu, chúng tôi thống nhất quy trình làm việc như sau:

- Nhánh chính: main để lưu phiên bản ổn định.
- Nhánh phát triển: develop làm trung tâm để merge tất cả nhánh con.
- Nhánh cá nhân: mỗi thành viên tạo feature-[tên] và test-[tên] để phát triển độc lập.
- Pull Request: khi hoàn thành tính năng, thành viên mở PR từ nhánh cá nhân sang develop.
- Review: một thành viên khác phải review/comment trước khi merge.
- Merge: trưởng nhóm Toàn thực hiện merge sau khi review.
- Release: dùng tag (v1.0, v1.1) để đánh dấu mốc quan trọng.

Quy trình này giúp chúng tôi làm việc song song mà không sợ ghi đè file của nhau, đồng thời đảm bảo mỗi thay đổi đều được xem xét trước khi đưa vào nhánh phát triển.

---

### 2. Các bước Git đã thực hiện

- Hoạt động 1 - Chuẩn bị môi trường  
Trưởng nhóm tạo repo GitHub GroupProject-Buoi3, thêm các collaborator.  
Sau đó tạo nhánh develop và đẩy lên remote, rồi mỗi thành viên tạo nhánh cá nhân (feature-Toan, feature-Thien, feature-Dung, cùng các nhánh test tương ứng).
- Hoạt động 2 - Commit cá nhân  
Mỗi thành viên tạo ít nhất 2 file (task-[tên], note-[tên]), thực hiện 2 lần commit đầu tiên.  
Sau đó thêm 1 file nữa (extra-[tên]) và commit thêm. Tổng cộng mỗi nhánh feature có tối thiểu 3 commit.  
Các commit được đặt tên rõ ràng: feat: thêm file..., chore: cập nhật ghi chú...
- Hoạt động 3 - Pull Request và Review  
Mỗi thành viên mở PR từ nhánh feature sang develop.  
Người khác review và thêm ít nhất 1 comment.  
Trưởng nhóm Toàn chịu trách nhiệm merge.  
Ngoài ra, chúng tôi tạo thêm 1 PR nhỏ để luyện tập thêm kỹ năng.
- Hoạt động 4 - Thực hành conflict  
Hai thành viên sửa cùng một dòng trong README.md để tạo xung đột.  
Khi merge vào develop sẽ phát sinh conflict, nhóm trưởng mở file, xóa marker <<<<<< =====>>>>>>, chọn nội dung đúng và commit lại.  
Lập lại với file task-shared.md để thực hành thêm.
- Hoạt động 5 - Quản lý tag  
Trưởng nhóm tạo tag v1.0 tại mốc ổn định đầu tiên.  
Sau một chỉnh sửa nhỏ, tạo tag v1.1 và push lên remote.  
Các thành viên khác clone repo mới, fetch tag và checkout v1.0, v1.1 để kiểm tra.
- Hoạt động 6 - Báo cáo nhóm  
Nhóm viết báo cáo này theo cấu trúc đề bài: quy trình, các bước đã làm, vấn đề và khắc phục, cảm nhận cá nhân và kết luận.
- Hoạt động 7 - Branch chung  
Trưởng nhóm tạo branch feature-shared.  
Mỗi người thêm file shared-[tên].txt và commit, sau đó tất cả cùng sửa group-note.md.  
Cuối cùng merge feature-shared vào develop.
- Hoạt động 8 - Rollback  
Dùng thực hành git reset --soft HEAD~1 để hủy commit nhưng giữ thay đổi.  
Thiên thực hành git revert để tạo commit đảo ngược an toàn.  
Toàn thực hành git restore để phục hồi file từ phiên bản trước.

---

### 3. Vấn đề và cách khắc phục

- Conflict khi merge: giải quyết bằng cách chỉnh sửa file, bỏ marker <<<<<<, =====, >>>>>>, chọn nội dung đúng rồi commit lại.
- Cảnh báo CRLF/LF: thêm file .gitattributes để chuẩn hóa, và cấu hình git config --global core.autocrlf input.
- Repo lồng repo: khi lỡ git init trong thư mục con, cần xóa .git thừa để tránh lỗi khi add và commit.
- Thiếu file khi push: sử dụng git status và git add --all để đảm bảo không bỏ sót.

---

### 4. Cảm nhận cá nhân

- Toàn: Học được quy trình teamwork với Git, hiểu rõ cách tạo nhánh, commit nhỏ, mở pull request và xử lý conflict. Thấy Git rất hữu ích khi làm việc nhóm.
- Thiên: (để bạn ấy bổ sung sau)
- Dũng: (để bạn ấy bổ sung sau)

---

### 5. Kết luận

So sánh làm việc có Git và không có Git:

- Khi có Git: dễ kiểm soát lịch sử thay đổi, dễ rollback nếu commit sai, có thể làm việc song song mà không ghi đè lẫn nhau. Code review qua pull request giúp đảm bảo chất lượng.
- Khi không có Git: mỗi người chỉnh sửa file trực tiếp, dễ bị ghi đè, khó quay lại phiên bản cũ, việc tổng hợp kết quả nhóm trở nên rối rắm và mất thời gian.

Nhờ áp dụng Git, nhóm chúng tôi thấy quá trình làm việc rõ ràng, minh bạch và hiệu quả hơn rất nhiều.