Git Workflow – Take Note

# 1. Git Workflow là gì?

Git Workflow là cách nhóm bạn tổ chức và phối hợp làm việc trong một dự án dùng Git – tức là: mỗi người làm việc trên nhánh nào, push code ở đâu, merge như thế nào, ai duyệt thay đổi, và quy trình đóng góp ra sao. Vì **Git là hệ phân tán**, bạn có nhiều cách tổ chức workflow khác nhau, không bị ràng buộc như SVN (Subversion).

# 2. Các mô hình Git Workflow phổ biến

## A. Centralized Workflow (Mô hình tập trung)

* Một repo trung tâm (trên GitHub/GitLab), giống SVN.
* Mỗi lập trình viên clone về làm trên máy → commit → push trực tiếp lên repo trung tâm.

**Đặc điểm:**

* Dễ hiểu, dễ triển khai, phù hợp nhóm nhỏ.
* Nhưng dễ bị **xung đột nếu nhiều người cùng sửa file** và push cùng lúc.

## B. Feature Branch Workflow

* Mỗi tính năng được phát triển trên một nhánh riêng (feature/login, feature/search)
* Sau khi làm xong, merge vào main hoặc develop (qua pull request)

**Đặc điểm:**

* Giúp tách biệt từng tính năng, dễ review và test.
* Phù hợp với team dùng CI/CD.

## C. Git Flow

* Tổ chức công việc theo nhiều loại nhánh: main, develop, feature/\*, release/\*, hotfix/\*
* Chuẩn cho dự án có vòng đời phát hành dài hạn.

**Đặc điểm:**

* Mạnh mẽ, rõ ràng nhưng hơi phức tạp cho nhóm nhỏ hoặc dự án đơn giản.

## D. Forking Workflow

* Phổ biến trên GitHub (dự án mã nguồn mở)
* Contributor **fork** repo → làm việc → push lên repo của họ → gửi **pull request** về repo gốc.

**Đặc điểm:**

* Ai cũng có thể đóng góp mà không cần quyền ghi.
* An toàn, dễ kiểm soát – phù hợp dự án cộng đồng.

## E. Integration-Manager Workflow

* Gần giống Forking nhưng dành cho tổ chức:
  + Dev có repo riêng (fork)
  + Maintainer review & merge pull request

**Đặc điểm:**

* Có người chịu trách nhiệm hợp nhất code
* Kiểm soát chất lượng dễ hơn

## F. Dictator and Lieutenants Workflow

* Dùng cho dự án cực lớn như Linux
* Gồm:
  + Các dev → gửi patch/PR cho các **Lieutenants**
  + Lieutenants merge → gửi tiếp cho **Dictator**
  + Dictator quyết định merge cuối cùng

**Đặc điểm:**

* Phân quyền rõ ràng
* Phù hợp tổ chức lớn, phân tán nhiều team

# 3. So sánh các Workflow phổ biến

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Workflow | Dễ dùng | Phức tạp | Phù hợp với ai? |
| Centralized | ✅ | ❌ | Nhóm nhỏ, mới học Git |
| Feature Branch | ✅ | ❌ | Nhóm trung bình đến lớn |
| Git Flow | ❌ | ✅ | Dự án chuyên nghiệp |
| Forking | ✅ | ❌ | Cộng đồng mã nguồn mở |
| Integration-Manager | ⚠️ | ⚠️ | Công ty với nhiều nhánh phụ |
| Dictator & Lieutenants | ❌ | ✅✅ | Dự án lớn nhiều người |

## 4. Các nguyên tắc chung trong mọi workflow:

* **Pull trước khi push** để tránh xung đột
* **Commit nhỏ gọn, rõ ràng**
* **Luôn test trước khi merge**
* **Dùng Pull Request để review code**
* **Tách nhánh khi làm từng tính năng mới**