# Kế Hoạch Triển Khai generate\_timeline\_rb\_v5.py (Tầng Sâu Nhất)

## I. Mục Tiêu

- Xây dựng module mới generate\_timeline\_rb\_v5.py để hỗ trợ multi-CC.  
- Giữ nguyên toàn bộ logic xử lý timeline từ phiên bản gốc timeline\_rb\_v4\_3\_1.py.  
- Bổ sung khả năng khởi tạo current\_time từ start\_time thay vì mặc định = 0.  
- Đảm bảo đầu vào – đầu ra hoàn toàn tương thích với pipeline gốc.

## II. Nguyên Tắc Thực Hiện

- ✅ Tuyệt đối không thay đổi logic xử lý bên trong các vòng lặp timeline.  
- ✅ Chỉ thêm tham số start\_time (mặc định = 0 nếu chưa truyền).  
- ✅ current\_time = start\_time (thay vì current\_time = 0).  
- ✅ Giữ nguyên cấu trúc biến: steps, timeline, current\_time, action\_dict, ...  
- ✅ Đảm bảo module mới có thể dùng thay thế module cũ mà không gây sai lệch timeline.

## III. Kế Hoạch Từng Bước

Bước 1: Sao chép file timeline\_rb\_v4\_3\_1.py thành generate\_timeline\_rb\_v5.py.  
Bước 2: Đổi tên hàm chính thành generate\_timeline\_rb\_v5(...).  
Bước 3: Bổ sung tham số start\_time với giá trị mặc định = 0.  
Bước 4: Thay đổi dòng khởi tạo current\_time = start\_time thay vì 0.  
Bước 5: Giữ nguyên toàn bộ logic xử lý còn lại: for step in steps, các vòng lặp, action\_dict, ...  
Bước 6: Kiểm tra lại toàn bộ đầu vào – đầu ra (args, return) khớp với phiên bản cũ.  
Bước 7: Ghi log hoặc tạo kiểm thử snapshot timeline để so sánh v4 vs v5 với start\_time = 0 (kết quả phải giống nhau).  
Bước 8: Kết nối vào pipeline cải tiến (tạm thay thế v4 trong run\_pipeline\_v5 để thử nghiệm).  
Bước 9: Tối ưu nếu cần (chuyển các giá trị tạm sang biến riêng, chuẩn hóa style...).

## IV. Kết Luận

Việc bắt đầu từ tầng sâu nhất của pipeline đảm bảo kiểm soát chặt chẽ luồng thời gian và tên gọi biến. Triển khai generate\_timeline\_rb\_v5.py là bước quan trọng để đảm bảo timeline giữa các CC được liên tục, trong khi vẫn giữ nguyên toàn bộ logic và kết quả xử lý của pipeline gốc.