

# 7. Quy trình phối hợp & quản lý thay đổi các nhóm

## 7.1. Mục tiêu chương

Chương này quy định cách phối hợp giữa nhiều đơn vị phát triển nhằm:

- Tránh xung đột giữa FE, BE và Database
- Đảm bảo tích hợp ổn định
- Giữ tính tương thích giữa các module
- Kiểm soát các thay đổi ảnh hưởng hệ thống

## 7.2. Khái niệm / phạm vi áp dụng

Áp dụng khi:

- Có nhiều đơn vị cùng phát triển
- FE và BE tách source
- Nhiều FE dùng chung BE
- BE tách thành nhiều service

Mọi thay đổi liên quan:

- API contract
- Entity/Database

- Auth
- Domain/CORS

đều phải thông qua Core Team.

## 7.3. Quy định chính

### Nguyên tắc chung

- Dùng contract-first (OpenAPI là nguồn chuẩn)
  - Mọi thay đổi API hoặc DB phải được kiểm soát
  - Mọi thay đổi breaking phải qua Core Team
- 

### Trường hợp FE và BE khác source, dùng chung DB

Rủi ro:

- Sai contract API
- Response làm FE vỡ
- Migration xung đột

Quy định:

- BE thay đổi API/DB phải update OpenAPI
- Mọi thay đổi DB phải có migration
- Staging dùng DB chung

Quy trình:

1. FE tạo ticket: API CONTRACT ISSUE
  2. BE/Core xác nhận
  3. Fix backward compatible hoặc tạo API v2
- 

## Trường hợp FE và BE dùng DB riêng khi dev

Quy định:

- Có DB baseline chung
- Staging dùng DB chung

Quy trình:

1. Nếu staging lỗi → so version migration
  2. Không cho merge nếu thiếu migration
- 

## Trường hợp nhiều FE dùng chung một BE

Quy định:

- Dùng IdP chung (WSO2/Keycloak)
- Mỗi FE là một OAuth client riêng
- Gọi API bằng JWT

Quy trình:

1. FE đăng ký client\_id
  2. Core cấu hình redirectUri, scope
  3. Core cấp quyền
- 

## Trường hợp FE chung source, BE nhiều service

Quy định:

- FE cấu hình baseUrl theo service
- Auth dùng token chung
- CORS do Core Team quản lý

Quy trình:

1. Thêm service → tạo Integration Checklist
  2. Core cấu hình CORS và token audience
- 

## Trường hợp thay đổi response làm FE vỡ

Quy định:

- Chỉ được thêm field (non-breaking)
- Không đổi kiểu dữ liệu
- Không rename field
- Breaking change phải tạo /api/v2

Quy trình:

1. Tạo BREAKING CHANGE REQUEST
  2. Core duyệt
  3. Thông báo FE
- 

## Trường hợp xung đột entity/model

Quy định:

- Entity dùng chung → coi là shared model
- Phải qua Data Model Change Request

Quy trình:

1. Partner đề xuất field + migration
  2. Core review
  3. Approve hoặc yêu cầu sửa
- 

## Trường hợp migration xung đột

Quy định:

- Không sửa migration đã merge
- Tạo migration mới để fix forward

Quy trình:

1. Rebase
  2. Tạo migration mới
  3. Test staging
- 

## Trường hợp seed/master data bị sửa

Quy định:

- Seed core do Core Team quản lý
- Partner chỉ được đề xuất

Quy trình:

1. Tạo MASTER DATA REQUEST
2. Core duyệt và triển khai

## 7.4. Quy trình triển khai tính năng nhiều tầng

1. Tạo ticket Feature
2. Chốt API contract (OpenAPI)
3. Chốt entity/migration
4. BE implement
5. FE integrate

6. Test staging

7. Release

## 7.5. Ví dụ minh họa

### 1. bảng mô tả các môi trường hệ thống:

Env	FE URL	BE URL	DB	Auth
dev	<a href="http://localhost:4200">http://localhost:4200</a>	<a href="http://localhost:8080">http://localhost:8080</a>	local	mock
staging	<a href="https://staging.app.vn">https://staging.app.vn</a> <a href="#">n</a>	<a href="https://staging.api.vn">https://staging.api.vn</a>	shared	WSO2 thật
prod	<a href="https://app.vn">https://app.vn</a>	<a href="https://api.vn">https://api.vn</a>	prod	WSO2 thật

Phiên bản #4

Được tạo 2026-02-23 07:50:12 UTC bởi Admin

Được cập nhật 2026-02-24 10:23:39 UTC bởi Nam Đặng