Microsoft Fabric — Three■Repo CI/CD Strategy (Expanded)

1) Purpose

Isolate Medallion layers into separate repos: fabric■bronze, fabric■silver, fabric■gold. Each has dev/uat/prod branches and its own pipeline.

2) Repo Layout

```
Repo: fabric-bronze (branches: dev, uat, prod)
Repo: fabric-silver (branches: dev, uat, prod)
Repo: fabric-gold (branches: dev, uat, prod)
```

3) Workspace ↔ Repo/Branch Map

Workspace	Repo	Branch	Folder
DEV_Bronze_WS	fabric-bronze	dev	/
UAT_Bronze_WS	fabric-bronze	uat	/
PROD_Bronze_WS	fabric-bronze	prod	/
DEV_Silver_WS	fabric-silver	dev	/
UAT_Silver_WS	fabric-silver	uat	/
PROD_Silver_WS	fabric-silver	prod	/
DEV_Gold_WS	fabric-gold	dev	/
UAT_Gold_WS	fabric-gold	uat	/
PROD_Gold_WS	fabric-gold	prod	/

4) PR & Branching Rules

• Work happens in the repo of the target layer. • Promotion path per repo: feature → dev → uat → prod. • Repo■level branch policies and CODEOWNERS ensure correct reviewers.

5) Pipelines (3 total — one per repo/layer)

```
trigger: { branches: [ dev, uat, prod ] }
pr: { branches: [ uat, prod ] }

stages:
- stage: Dev
  jobs:
  - job: DeployBronzeDev
    steps:
    - bash: python deploy_fabric.py --layer bronze --workspace $(DEV_BRONZE_WS_ID) --branch dev
```

6) Real■World Example — Parallel Teams

```
Alice (fabric-bronze):
  feature → PR→dev → Bronze Dev stage → PR dev→uat → Bronze UAT stage → PR uat→prod → Bronze Pr
Bob (fabric-silver):
  feature → PR→dev → Silver Dev stage → PR dev→uat → Silver UAT stage → PR uat→prod → Silver Pr
Carol (fabric-gold):
  feature → PR→dev → Gold Dev stage → PR dev→uat → Gold UAT stage → PR uat→prod → Gold Prod sta
```