

Version 2026.1.0

# 开放教学系统 探索工作坊

从内容交付走向系统运行的范式转移

## 核心定位

以 **Course as Code (课程即代码)** 为方法论内核，将课程视为一个可被设计、运行与治理的系统，而非静态的教学内容集合。

### BEFORE

#### 传统模式

- × 依赖个人经验
- × 静态内容交付
- × 难以规模化协作



### TARGET

#### 开放系统

- ✓ 结构化运行
- ✓ 持续演进 (Git)
- ✓ 教学资产沉淀



## 工作坊三期路线图

### PHASE 01 · 认知重构

#### 为什么课程必须被系统化？

解决“为什么要变”的问题。建立对“教学系统问题”的共同认知，从教学内容认知转向教学系统设计。

产出：共识与问题背景

### PHASE 02 · 结构设计

#### Course as Code 如何成立？

解决“设计成什么样”的问题。理解单一信任源、变更追溯与评审机制。

结构设计

工程化思维

### PHASE 03 · 运行实践

#### 教学如何与 Git 协同？

解决“怎么跑起来”的问题。通过 Training Sandbox 演示任务发布、PR提交与Review治理。

产出：行动准备

## Training Sandbox

最小运行示例 (Minimal Runnable Example)



```
training-sandbox/tree

training-sandbox/
├── README.md           # 入口说明
├── COURSEFILE.yaml     # 课程声明
├── tasks/              # 任务库(源代码)
│   ├── T01-issue.md
│   └── T02-pr-submit.md
├── templates/         # 规范模板
│   └── review-rubric.md
└── examples/          # 成果示例
```

### 核心价值

- 不是复杂的课程主仓，而是“看一眼就懂”的结构演示。
- 展示教学闭环：任务发布 → PR提交 → 教师Review → 沉淀。

## 试点推进策略：低风险，MVP



### 局部切入

不重构整门课，只做一个实验单元或项目。



### 角色协同

Maintainer 控节奏，Pilot 教师做教学。



### 过程沉淀

重视失败经验与边界条件的记录。

## Key Takeaway

开源不仅是内容，更是一种职业核心能力。

我们不是在教授Git工具，而是在构建支撑协作、演进与治理的开放教学系统。