

Week06 | 课时 5 | Data Factory Runbook: 从个人脚本到团队可交接系统

Table of contents

Runbook 不是事后文档，而是数据工厂的操作接口	1
这节课解决什么问题	1
参考学习时间	2
学完这一讲，你应该能做到什么	2
本课产出	2
先看故障恢复泳道图	3
1. Runbook 最小结构	3
2. UI 与 CLI 两条路径	4
3. Recovery decision tree	5
4. 交接给后续周次的字段	7
5. 已知限制必须写清	7
优秀 runbook vs 不合格 runbook	7
自检清单	8
课后最小行动	8

Runbook 不是事后文档，而是数据工厂的操作接口

这一讲把 Week06 收口成团队能照着执行的 Data Factory Runbook。

[进入实验](#) [返回 Week06 总览](#)

下载讲义

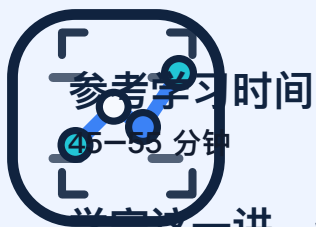
提供适合离线阅读的 PDF 版和适合批注整理的 Word 版。

[PDF 版 · 打印 / 离线阅读](#) [Word 版 · 批注 / 二次整理](#)

这节课解决什么问题

一个数据工厂是否可运营，不取决于作者本人能不能救火，而取决于：

另一个同学拿到 runbook 后，能不能定位、决策、重跑、验证、记录和交接。



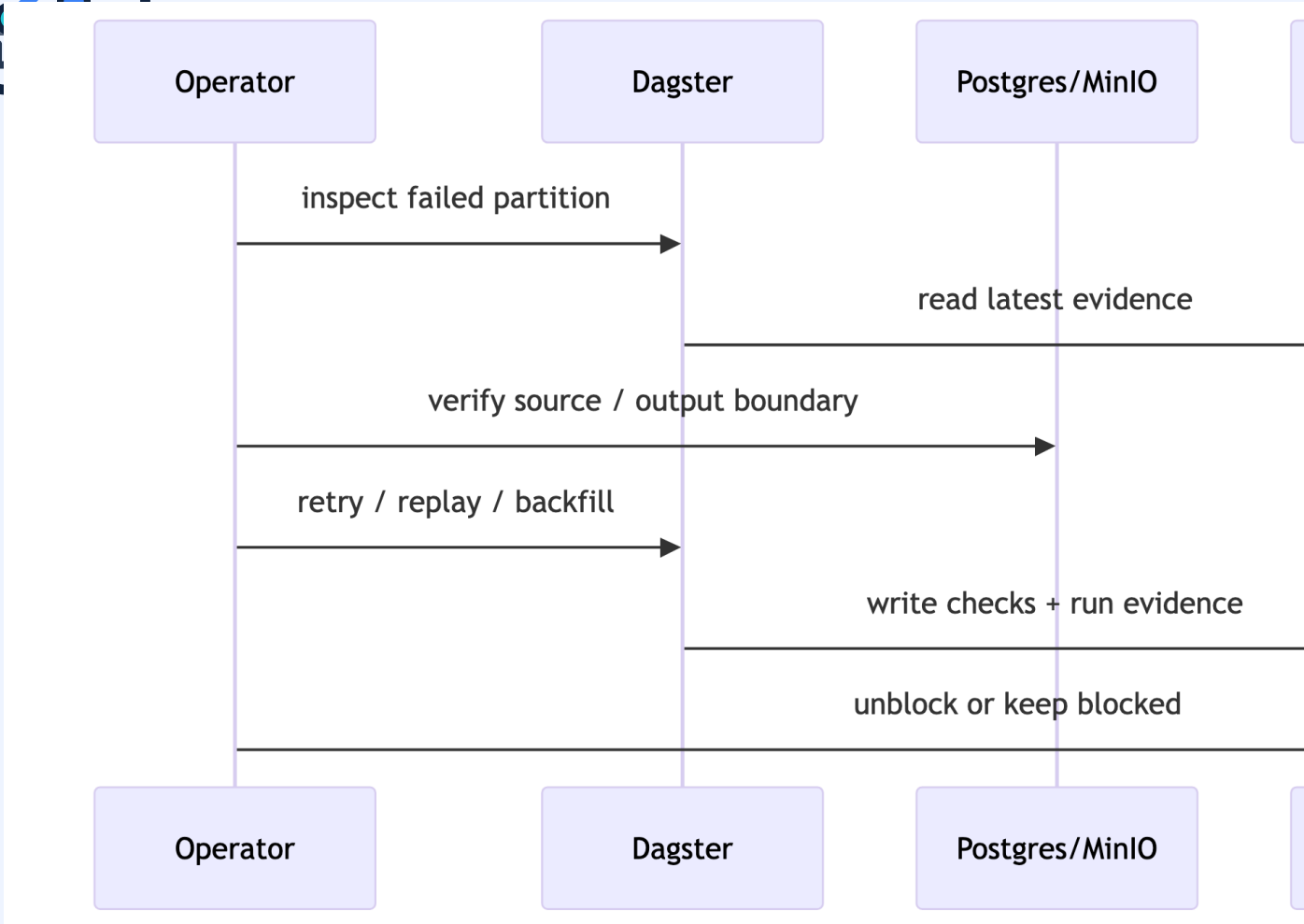
学完这一讲，你应该能做到什么

1. 写出 Week06 Data Factory Runbook 的最小结构。
2. 设计 UI 与 CLI 两条恢复路径。
3. 用 reason code 记录为什么 retry / replay / backfill。
4. 把 run evidence 交接给 Week07 / Week08 / Week11 / Week14。
5. 说明已知限制，而不是伪装生产完整。

本课产出

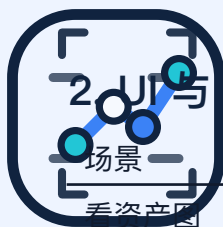
- [runbooks/week06-data-factory.md](#)
- [reports/week06/week06_delivery_summary.md](#)

先看故障恢复泳道图



1. Runbook 最小结构

```
runbooks/week06-data-factory.md
1. Scope and non-goals
2. Required services
3. Environment variables
4. Load Definitions
5. Observe sources
6. Materialize core assets
7. Backfill / replay decision tree
8. Run checks
9. Read evidence reports
10. Known limitations and handoff
```



2. UI 与 CLI 两条路径

	UI 路径	CLI 路径
	Dagster Asset Graph	Definitions.validate_loadable(defs) smoke test
补某个分区	选择 partition materialize / backfill	python -m pipelines.data_factory.backfill_plan --partition 2026-04-17 --mode dry-run
看检查结果	Asset Checks tab	pytest tests/integration/test_week06_asset_checks.py -q
看证据	Asset materialization metadata	reports/week06/run_evidence/*.json

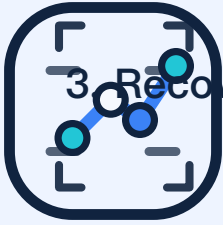
课程里两条路都要讲，因为团队协作不能只依赖 UI，也不能只依赖命令。

真实课堂命令以 runbook 为准：

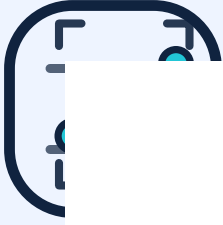
```
# 可执行: 已与项目 runbook 对齐
docker compose --profile tools --env-file infra/env/.env.local -f infra/docker-compose.yml run --rm devbox \
  pytest tests/contract/test_week06_run_evidence_schema.py -q

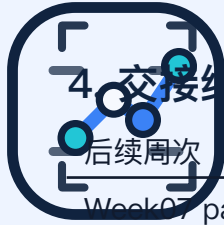
docker compose --profile tools --env-file infra/env/.env.local -f infra/docker-compose.yml run --rm devbox \
  pytest tests/integration/test_week06_asset_graph_smoke.py -q
```

验收点：UI 能帮助定位，但最终交接要落到 reports/week06/ 和 runbooks/week06-data-factory.md。



3 Recovery decision tree





4. 交接给后续周次的字段

后续周次	Week06 要交接什么
Week07 parse	source fingerprint、 doc version、 partition_key、 manifest_id
Week08 retrieval	baseline asset、run evidence、release_id、 trace_id
Week11 evals	eval_run_id 预留、 data release、 quality gate
Week14 governance	lineage snapshot、git sha、reason codes、 decision log

5. 已知限制必须写清

Runbook 里必须承认:

- Week04 lakehouse 未完成时, snapshot 字段可以为空
- Week05 analytics 未完成时, semantic/mart 状态可以 skipped
- Student Core 不依赖 Dagster+
- OpenLineage / lakeFS / OpenMetadata 留给 Week14
- Week07 解析、Week08 索引不在 Week06 实现

优秀 runbook vs 不合格 runbook

维度	优秀 runbook	不合格 runbook
前置条件	写清服务、env、默认 partition、可选依赖状态	只说“先启动环境”
故障判断	用 reason code 区分缺数据、schema、重复、optional unavailable	统一写“重跑一下”
操作路径	UI 路径和 CLI 路径都能走	只贴截图或只贴命令
验收证据	指向 checks、run evidence、delivery summary	只有日志片段
交接	写清 downstream unblock / hold 决策	作者本人能懂, 别人无法接手



! 核心判断

没有 runbook 的数据工厂不可交接；不能交接，就不能算工程完成。

自检清单

- runbook 能让别人启动、观察、materialize、backfill、检查和读 evidence
- recovery decision tree 能区分 retry / replay / backfill / restore
- optional 依赖的限制写清楚
- evidence 能交给 Week07 / Week08 / Week11 / Week14
- 没有把 Week06 写成 UI 点击教程

课后最小行动

为你的 Week06 runbook 补一个 operator checklist:

```
## Operator checklist

- [ ] 当前失败 partition:
- [ ] 已读取 latest evidence:
- [ ] 已确认 source / output boundary:
- [ ] 选择动作: retry / rerun / replay / backfill / restore
- [ ] checks 结果:
- [ ] downstream decision:
```