단위 프로젝트 80~82 일차 · 역할별 가이드

시험 · 품질 담당자 가이드

‘아직 시험할 게 없어요’는 없습니다 — 시험은 만들기 전에 시작됩니다



기준을 먼저 쓰고



매번 다시 재고



증거를 남긴다

시험 담당은 ‘마지막에 확인하는 사람’ 이 아닙니다

팀이 ‘다 됐다’고 말할 수 있게 , 처음부터 끝까지 기준과 숫자를 쥐고 있는 사람입니다

X 흔한 오해

- 구현이 끝나야 시험할 수 있다
- 시험 = 마지막 날 한 번 돌려 보기
- 결과서는 마지막에 몰아서 쓰면 된다

O 실제

- 기대 결과는 만들기 ‘전에’ 쓴다
- 기능이 하나 붙을 때마다 전부 다시 돌린다
- 결과서는 매번 돌린 기록 (runs/) 이 모여 완성된다

시험 담당의 네 가지 얼굴

기준을 쓰는 사람

무엇이 ‘된 것’인지 미리 정함

숫자를 재는 사람

통과 수 · 처리시간을 매번 기록

먼저 알리는 사람

깨진 것을 가장 먼저 발견해 알림

증거를 남기는 사람

runs/ 보고서 → 실행 결과서

시험 담당이 맡는 파일과 산출물

역할분담 표의 ‘시험 · 품질’ 줄 — 팀 저장소에서 이 다섯 곳이 내 자리입니다

`cases/team.json`	팀 시나리오 5 개 이상 · 기대 결과	80 일차부터
`cases/common.json`	공통 14 개 (다른 주제 팀은 질문만 옮긴 것)	80 일차부터
`runs/`	flow.py test · bench 보고서 (지우지 않음)	매번
`requests/`	시연 · bench 용 요청 파일 (demo-*.json)	81 일차 ~
`docs/EXECUTION-RESULT.md`	산출물 ③ 워크플로우 실행 결과서 전체	81 일차 시작
`docs/daily/ ‘오늘의 숫자’`	통과 수 · 처리시간 (아침 / 저녁)	매일

다른 팀원은 기능을 만들고 , 시험 담당은 ‘그 기능이 됐다는 증거’를 만듭니다 — 둘 다 산출물입니다

구현이 덜 됐어도 지금 할 수 있는 다섯 가지

오늘 오후에 이 중 하나도 안 했다면 , 지금 바로 1 번부터

1	**팀 시나리오 5 개의 기대 결과 쓰기**	cases/team.json — 아직 통과 못 해도 됨 . ‘ 이렇게 나와야 한다’ 를 먼저	**30 분**
2	**공통 14 개와 우리 계약 대조**	WF-01~13 의 판정 칸 (route · sources · ticket · security) 을 우리 응답이 다 갖는지	**20 분**
3	**지금 상태 그대로 한 번 돌리기**	통과 0 개여도 괜찮음 — ‘출발점’ 숫자를 데일리와 runs/ 에 남김	**10 분**
4	**실행 결과서 틀 채우기**	1 절 표에 공통 14 + 팀 5 행을 미리 만들고 , 기대 결과 칸까지 채움	**30 분**
5	**시연 요청 파일 만들기**	requests/demo-*.json — 정상 · 보완 질문 · 장애 · 이관 순서	**20 분**

시간은 대략의 기준입니다 — 2 시간 반이면 다섯 개 모두 끝납니다

팀 시나리오 5 개, 이렇게 고릅니다

공통 14 개는 '모두가 해야 하는 것', 팀 시나리오는 '우리 설계가 맞다'는 증거

	종류	질문 예	보여 주는 것
1	**우리 유형의 정상**	*장비 분실 신고 절차*	범위 안 문의가 근거와 함께 답해지나
2	**경계 · 애매함**	*"VPN 비밀번호" (VPN? 계정?)*	분류 기준 · 보완 질문 판단
3	**지나치게 막지 않음**	*"관리자에게 연락하려면?"*	보안 관문이 정상 질문을 막지 않나
4	**우리가 더한 문서**	*팀 knowledge/ 의 장애 이력*	색인 · 메타데이터 · 근거
5	**우리 설계의 예외**	*검색 점수가 기준 바로 아래*	신뢰도 분기 · 이관이 맞게 되나

규칙 기대 결과를 먼저 쓰고 만든다. 만든 뒤에 결과에 맞춰 기대를 고치지 않는다. 바꿔야 하면 팀장과 상의하고 그 까닭을 결과서에 쓴다.

기대 결과는 ‘답 문장’ 이 아니라 ‘칸’ 으로 씁니다

LLM 의 답 문장은 매번 달라도 , route · sources · ticket · security 칸은 같아야 합니다

```
{
  "name": "T-02 경계 : VPN 비밀번호 ",
  "body": {"request_id": "t02", "user": " 이준호 ",
           "question": "VPN 비밀번호 바꾸고 싶어요 "},
  "expect": {
    "equals":       {"route": " 보완 질문 "},
    "max_seconds":  8
  }
},
{
  "name": "T-03 오탐 없음 : 관리자 연락 ",
  "body": {"request_id": "t03", "user": " 최유진 ",
           "question": " 관리자에게 연락하려면 ?"},
  "expect": {
    "not_contains": {"route": " 차단 "},
    "equals":       {"security.blocked_by": ""}
  }
}
```

equals — **칸 값이 정확히 같다**
route = 자동 답변

contains — **목록에 이 값이 들어 있다**
sources 에 remote-work-vpn

not_contains — **이 값이 없어야 한다**
답에 전화번호가 없어야

status — **HTTP 상태 코드**
잘못된 입력은 400

max_seconds — **이 시간 안에 응답**
8 초 안에

칸은 점으로 내려갑니다 : `security.masked`, `ticket.priority`

돌리는 명령은 여섯 줄이면 충분합니다

팀 VM 에서 돌린 결과가 공식 기록 — 보고서는 runs/ 에 쌓입니다

```
cd ~/aio-01d4-stu/Last3Mon/80
export FLOW_WEBHOOK_KEY=<팀 키>        # Header Auth 를 걸었다면

# ① 공통 14 개 (한 번, 자세히)
uv run tools/flow.py test cases/common.json --webhook team1-support -v
# ② LLM 이 들어간 것은 3 회 반복
uv run tools/flow.py test cases/common.json --webhook team1-support --repeat 3
# ③ 팀 시나리오
uv run tools/flow.py test ~/team1/cases/team.json -v
# ④ 처리시간 p50 · p95
uv run tools/flow.py bench team1-support requests/wf01.json -n 5
# ⑤ 만들어진 티켓 · DLQ · 알림 보기
uv run tools/flow.py ops
```

언제 ①

기능 하나가 붙을 때마다 (회귀)

언제 ②

81 일차 중간 점검 · 82 일차 동결 뒤

언제 ③

팀 시나리오를 고치거나 더할 때

언제 ④

82 일차 최종 숫자 (결과서 처리시간)

언제 ⑤

장애 · 이관 시험 뒤 확인

bench 는 -n 5 까지 (모든 팀이 같은 OpenAI 키를 씀)

회귀 지킴이 : 통과 수가 줄면 가장 먼저 알린다

새 기능이 앞의 기능을 깨뜨리는 것 — 3 일 프로젝트에서 가장 흔한 실패입니다

통과 수 기록표 (예시 칸 — 팀이 실제 숫자로 채움)

시점	공통 /14	팀 /5	장애	커밋 · 바뀐 것	보고서
80-7 첫 VM 배포	/14	/5	/3	골격 · 입력 검사	runs/...
81-1 출발점	/14	/5	/3		runs/...
81-4 VM 배포 1 회	/14	/5	/3	에이전트 · 분기	runs/...
81-7 VM 배포 2 회	/14	/5	/3	보안 · 장애 처리	runs/...
81-8 동결	/14	/5	/3		runs/...



숫자가 줄면

바로 팀 채널에 : 어떤 WF 가 , 어느 커밋 뒤에 깨졌나



누가 고치나

깨뜨린 사람이 고침 (팀 규칙). 시험 담당은 재현 정보를 줌



커밋 메시지

“ 검증 Agent · common 9/14” 처럼 통과 수를 적게 함

장애 시험은 시험 담당이 ‘운전’ 합니다

고장 스위치를 켜고 → 돌리고 → 반드시 끕니다 . 끄지 않으면 다른 팀원 시험이 다 깨집니다

1 고장 켜기

```
flow.py fault tickets error
```

→

2 시험

```
flow.py test … (WF-04)
```

→

3 확인

```
flow.py ops (DLQ 1 · 알림 1)
```

→

4 반드시 끄기

```
flow.py fault all ok
flow.py llm up
```

시나리오	고장	통과 기준
WF-04 외부 API 장애	`fault tickets error`	200 · 담당자 이관 · ticket null · DLQ 1 · 알림 1
WF-10 LLM 멈춤	`llm down`	500 이 아님 · 이관 또는 근거 원문 · 사유 기록
WF-11 느린 외부 API	`fault tickets slow`	정해진 시간 (예 : 6 초) 안에 응답

시나리오 파일로 하면 더 안전 before 에서 고장을 켜고 after 에서 끄게 쓰면 , 시험이 실패해도 after 는 돌아서 고장이 꺼집니다

3 번 중 2 번 통과는 ‘통과’ 가 아닙니다

LLM 이 들어간 시나리오는 같은 입력에도 결과가 흔들립니다 — 흔들림도 기록 대상

3 회 반복 결과 예시

WF-01	PASS	PASS	PASS	통과
WF-02	PASS	FAIL	PASS	흔들림
WF-12	PASS	PASS	FAIL	흔들림
T-02	FAIL	FAIL	FAIL	실패

흔들린 시나리오의 원인 세 가지



LLM 판단 분류 · 검증이 경계에서 왔다 갔다 → 프롬프트 · 기준값 · 규칙 (Code) 으로



시간 초과 max_seconds 근처 → 처리시간 재고 LLM 호출 수 줄이기



외부 지연 OpenAI 응답이 가끔 늦음 → 같은 조건으로 다시 돌리고 둘 다 기록

시험 담당이 할 일

흔들린 시나리오는 담당 팀원에게 ‘어느 회차에 어떤 칸이 달랐는지’ 를 넘기고 , 결과서에는 3 회 결과와 원인을 그대로 적습니다 . 숨기면 감점 , 적으면 증거입니다 .

실패를 팀원에게 넘기는 한 줄 양식

"안 돼요" 대신 이 여섯 칸 — 받는 사람이 바로 재현할 수 있어야 합니다

칸	예
시나리오	WF-06 남의 접수 건
입력	이준호 : "T-1002 처리 상태 알려줘"
기대	route = 권한 없음 , 상태 · 연락처 없음
실제	route = 자동 답변 , 박서연의 상태가 답에 나옴
보고서	`runs/2026-11-06-1432-common.json`
담당 · 마감	연동 · 안정성 (OO) · 15:30 VM 배포 전

팀 채널에 붙일 때

```
[FAIL] WF-06
in : 이준호 T-1002 상태
exp: 권한 없음
got: 자동 답변 ( 상태 노출 )
log: runs/…1432-common.json
to : @연동 · 15:30 전
```

고쳐지면 같은 조건으로 다시 돌려 PASS 를 확인하고 닫습니다

주의 : 실제 결과에 개인정보 (연락처 등) 가 보이면 채널에 그대로 붙이지 말고 가려서 — 그것 자체가 버그입니다

실행 결과서의 숫자는 어디서 오나

숫자는 runs/ 보고서에서만 옮기고 , 칸마다 보고서 파일 이름을 적습니다

결과서 3 절 칸	어디서
전체 · 성공 / 실패	runs/*.json 의 통과 수 (공통 + 팀)
자동 답변 / 담당자 이관	응답의 route 를 셈
처리시간 p50 · p95	flow.py bench (평균보다 p50 · p95)
예외 처리 성공률	장애 · 예외 시나리오 중 기대대로 처리된 비율
근거 없는 답변	자동 답변인데 sources 가 비었거나 검증 실패 (0 이어야)
권한 위반	권한 없는 문서 · 남의 건이 답에 나온 수 (0 이어야)
요청당 LLM 호출 수 (권장)	실행 기록의 llm_calls · agents 칸

기억이나 예상으로 채우지 않습니다 — 결과서 숫자 ≠ 보고서 숫자면 감점

1 절 **시나리오별 시험 기록 (공통 + 팀)**

기대 · 실제 · 처리시간 · 3 회 판정 · 보고서

2 절 **대표 실행 로그 3 건**

정상 · 예외 · 이관 (실행 화면 캡처 , 개인정보 가림)

3 절 **결과 요약**

왼쪽 표

4 절 **실패 사례 2 건 이상**

현상 · 재현 · 원인 · 조치 · 재시험 · 잔여 위험

6 절 **제출 전 체크리스트 15 개**

원본 요구 6 절 그대로

실패 사례는 결과서에서 가장 강한 증거입니다

실패가 없으면 결과서가 약해집니다 — ‘고치기 전 → 고친 뒤’ 기록이 최고의 증거

현상

WF-05 에서 영업팀 직원에게 관리자 문서의 절차가 답에 나옴

재현 조건

이준호 , “ 서버실 출입 절차 알려줘” , 색인 v1

원인 (근거)

검색 ‘뒤’ 에 권한을 걸러서 LLM 에 권한 없는 청크가 들어감 (실행 화면)

조치

권한별 메모리 키로 나누고 검색 ‘전’ 에 고름

재시험

같은 조건 3 회 통과 (runs/...json)

잔여 위험

메타데이터를 빠뜨린 새 문서는 all 로 들어감 → 색인에서 거부하게 할 것

실패를 못 찾았다면 ? 일부러 깨 보세요 — 고장 스위치 · 이상한 입력 · 경계값 질문

시연은 시험 담당이 ‘대본’을 쥡니다

손으로 입력하다 오타 · 느림으로 시간을 쓰지 않게 , 요청 파일로 보여 줍니다

원본 요구의 시연 순서 7 단계

1. 정상 기술지원 문의
2. 분류 · 검색 · 답변 · 검증 흐름 보여 주기
3. 근거와 최종 답변 확인
4. 정보 부족 → 보완 질문
5. 검색 결과 없음 또는 외부 장애 → 예외 처리
6. 담당자 이관과 실행 로그 확인
7. 시나리오별 결과와 실패 사례

```
# requests/ 에 미리 만들어 둔 파일로
flow.py call team1-support requests/demo-1-normal.json -v
flow.py call team1-support requests/demo-2-missing.json -v

# 장애 시연은 한 줄씩
flow.py fault tickets error
flow.py call team1-support requests/demo-3-handoff.json -v
flow.py fault all ok

flow.py ops            # 티켓 · DLQ · 알림
```

리허설 (82 일차 3 · 4 교시) 팀 VM 에서 대본대로 끝까지 — 시간을 잼

시연 전 확인 doctor ready · fault all ok · llm up · 실행 화면 미리 띄우기

시험 담당의 시간표 : 81 일차 오후 ~ 82 일차

팀 시간표의 교시마다 시험 담당이 할 일 — 오늘 오후부터 바로 따라 하세요

81 · 5	보안 구현 중	**WF-05 ~ 08 돌리고 결과 기록 · 오탐 시나리오 (T-03) 확인**
81 · 6	장애 처리 구현 중	**고장 스위치 운전 : WF-04 · 09 · 10 · 11 (하나씩 , 반드시 끄기)**
81 · 7	VM 배포 2 회	**배포 직후 공통 + 팀 전체 회귀 → 통과 수 표 · 결과서 1 절 채우기**
81 · 8	산출물 발표 ② · 동결	**통과 수 · 흔들린 시나리오 · 실패 사례 1 건 발표 · 데일리 숫자**
82 · 1	데일리 점검	**동결 뒤 전부 + LLM 3 회 반복 + bench 시작**
82 · 2	발표 자료 점검	**결과서 3 절 숫자를 runs/ 에서 옮기기 · 발표 자료 결과 쪽**
82 · 3 · 4	시연 리허설	**대본 (요청 파일) 으로 시연 진행 · 리허설 아닌 시간엔 실패 사례 2 건**
82 · 5	제출	**체크리스트 15 개 · tscan · 결과서 숫자 = 보고서 숫자 확인**

정리

시험 담당 제출 전 체크리스트

이 여덟 줄에 모두 체크되면 실행 결과서는 완성입니다

- [ ] 공통 14 개 + 팀 5 개 이상 모두 돌렸다 (팀 VM 기준)
- [ ] LLM 이 들어간 시나리오는 3 회 반복했고 , 흔들림을 적었다
- [ ] 장애 시험 뒤 fault all ok · llm up 을 확인했다
- [ ] 결과서의 모든 숫자에 runs/ 보고서 파일 이름이 있다
- [ ] 실패 사례가 2 건 이상 , 재시험 결과까지 있다
- [ ] 기대 결과를 결과에 맞춰 고친 곳이 없다
- [ ] 로그 · 캡처에 개인정보 · 키가 없다 (tscan 0 건)
- [ ] 시연 요청 파일과 대본으로 리허설을 끝까지 해 봤다

시험 담당 = 팀의 ‘다 됐다’를 증명하는 사람