단위 프로젝트 80~82 일차 · 4 팀 공통

역할별 할 일 · 하지 말아야 할 일

팀장 · RAG · 워크플로우 · 통합 · 시험 / 품질 — 각자 무엇을 쥐고, 무엇을 놓아야 하나


팀장


RAG 담당


워크플로우 담당


통합 담당


시험 · 품질 담당

다섯 역할 , 한 장으로

가운데의 공통 계약 · 팀 저장소 · 팀 VM 을 다섯 역할이 함께 지킵니다

팀장

범위 · 계약 · 일정을 지키고 결정하는 사람

헌장 · README · AGENTS.md · 데일리
산출물 발표 총괄

워크플로우 담당이 2 명 이상이면 파일 (하위 워크플로우) 로 나눕니다 → 10 쪽

RAG 담당

문서를 근거로 바꾸는 검색 설계자

index · search 워크플로우 · knowledge/
산출물 ② RAG 설계서

공통 계약
팀 저장소 (main)
팀 배포 VM

워크플로우 담당

에이전트와 흐름을 만드는 사람

support(본체) · 에이전트 프롬프트
산출물 ① 설계서 1~4 · 6 절

시험 · 품질 담당

‘ 다 됐다’ 를 숫자와 증거로 증명하는 사람

cases/ · runs/ · requests/
산출물 ③ 실행 결과서

통합 담당

조각을 잇고 바깥과 연결해 VM 에 올리는 사람

handoff · errors · 어댑터 · 배포
산출물 ① 설계서 5 · 7 절

누가 결정하고 , 누가 실행하나

충돌이 생기는 대부분은 ‘이건 누가 정하는 거지 ?’ 에서 시작합니다

일	팀장	RAG	워크플로우	통합	시험 · 품질
범위 · 주제 · 조합 바꾸기	결정	의견	의견	의견	의견
공통 계약 (요청 · 응답 칸) 바꾸기	결정		실행	실행	확인
에이전트 구성 · 분기 기준	확인	의견	결정	의견	의견
청킹 · top_k · 점수 기준	확인	결정	의견		확인
main 병합	결정			실행	확인
팀 VM 배포 · 게시	결정			실행	확인
기대 결과 (시나리오) 바꾸기	결정	의견	의견	의견	실행
기능 동결 · 발표 순서	결정	의견	의견	의견	의견

결정 = 최종 판단 · 실행 = 직접 손으로 · 의견 = 결정 전에 물어봄 · 확인 = 바뀐 뒤 시험으로 확인

팀장 : 해야 할 것 · 하지 말아야 할 것

가장 많이 만드는 사람이 아니라 , 매일 ‘돌아가는 하나’와 ‘좁은 범위’를 지키는 사람

해야 할 것

- 범위를 좁게 지키기 — 유형 4~6 개 , ‘ 하지 않을 것’ 을 헌장에 적기
- 매일 아침 목표 숫자 정하기 (예 : 오늘 저녁 공통 11/14)
- 공통 계약을 지키기 — 바꿀 땐 설계서 3 절부터 고치고 공지
- 막힌 사람 챙기기 — 30 분 막히면 팀 , 1 시간이면 강사
- 결정은 기록 — 헌장 6 절 ‘바뀐 것’에 날짜 · 까닭
- 81 일차 8 교시 기능 동결 선언 , 발표 리허설 주관

하지 말아야 할 것

- 혼자 다 만들기 (막히면 역할을 다시 나눔)
- 남이 맡은 파일을 말없이 고치기
- ‘ 거의 다 됐어요’ 를 믿기 — 통과 수로만 판단
- 마감 직전에 새 기능 붙이기
- 시험 담당을 기다리게 두기 — 기대 결과부터 쓰게 하기

맡는 파일 · 산출물

docs/TOPIC · TEAM-CHARTER · daily/
README.md · AGENTS.md · 발표 자료

81 · 82 일차 핵심 순간

81-8 산출물 발표 ② · 동결 선언
82-1 · 2 시연 · 자료 점검 , 82-5 제출 확인

다른 역할에 넘겨줄 것

모두에게 : 오늘 목표 숫자 · 결정 사항
통합에게 : 병합 · 배포 승인

RAG 담당 : 해야 할 것 · 하지 말아야 할 것

‘검색이 된다’가 아니라 ‘왜 이 설정인지 숫자로 말할 수 있다’가 목표

해야 할 것

- 문서마다 권한 (access_level) · 절 · 출처 메타데이터 붙이기
- 청킹을 두 가지 이상 같은 질문으로 비교 (청크 수 · 길이 분포 · hit@3)
- 임베딩 모델 · 벡터 차원 · 거리 방식 기록
- 점수 기준은 정상 · 결과 없음 질문의 점수 분포를 보고 정하기
- 권한 필터는 검색 ‘전에’ 걸기 (WF-05)
- 색인은 index 워크플로우로 — 언제든 다시 만들 수 있게 , 시간 재기

하지 말아야 할 것

- 한 가지만 해 보고 근거 없이 결정
- 예시 값 (500 토큰 · 오버랩 80) 을 그대로 쓰기
- 검색 ‘뒤’에 권한 거르기
- n8n 화면에서 손으로 색인하고 기록 안 남기기
- 개인정보가 든 원문을 가리지 않고 색인

맡는 파일 · 산출물

workflows/< 팀 >-index · < 팀 >-search
knowledge/ · docs/RAG-ARCHITECTURE.md

81 · 82 일차 핵심 순간

80-6 색인 · 청크 분포
81-4 산출물 중간점검 (RAG 3~5 절)

다른 역할에 넘겨줄 것

워크플로우에게 : search 웹훅 · evidence 모양 · 점수 기준
시험에게 : 경계값 질문 · 우리가 더한 문서

워크플로우 담당 : 해야 할 것 · 하지 말아야 할 것

에이전트에게 역할을 나눠 주고 , AI 가 판단할 곳과 규칙이 판단할 곳을 가르는 사람

해야 할 것

- 역할 카드대로 Agent 분리 (분류 · 답변 · 검증 , 하지 않는 것까지)
- Agent 출력은 JSON → Code 가 읽고 , 못 읽으면 안전한 쪽 (이관) 으로
- 판단이 분명한 것 (권한 · 길이 · 중복 · 점수) 은 Code · If 로
- 분기 · 병렬 · 루프 각각 한 곳 이상 + 루프 최대 횟수
- 연결선 페이로드 표는 n8n 실행 화면에서 복사해 채우기
- 고칠 때마다 export → 저장소 커밋 (통과 수를 메시지에)

하지 말아야 할 것

- 한 Agent 에게 분류 · 검색 · 답변 · 검증을 다 시키기
- 같은 워크플로우 JSON 을 둘이 동시에 고치기
- 공통 계약의 칸 이름 · route 값을 마음대로 바꾸기
- 키 · 토큰을 노드나 JSON 에 직접 쓰기 (Credential 만)
- LLM 이 꼭 필요 없는 곳에 LLM 호출 넣기 (느리고 비쌈)

맡는 파일 · 산출물

workflows/< 팀 >-support (본체)
① 설계서 1~4 · 6 절

81 · 82 일차 핵심 순간

81-2 · 3 에이전트 · 이관 (WF-01 · 03)
81-4 루프 · 병렬 · 복합 (WF-02 · 11 · 12)

다른 역할에 넘겨줄 것

통합에게 : 이관 조건 · 하위 호출 지점
시험에게 : 바꾼 노드 · 기대에 영향 주는 변경

통합 담당 : 해야 할 것 · 하지 말아야 할 것

조각과 조각 , 우리와 바깥을 잇고 , 팀 VM 에 '돌아가는 하나' 를 올리는 사람

해야 할 것

- 하위 워크플로우 호출에 timeout · 오류 출력 달기
- 도구 노드 · Credential 정리 — 401 · 403 은 재시도 없이 중단 + 알림
- Error Workflow · DLQ · 알림 · 중복 방지 (request_id)
- B · C 조합이면 입구 · 뒷단 어댑터 (계약 모양으로 변환)
- VM 배포 실행 — 하위 먼저 게시 , 본체는 마지막 , 배포 뒤 회귀 요청
- VM 의 Credential 이름 · ID 를 팀 노트북과 똑같이 맞추기

하지 말아야 할 것

- VM 의 n8n 화면에서 직접 고치기 (저장소가 정답)
- 컨테이너 사이 주소에 localhost 쓰기 (서비스 이름 · 안쪽 포트)
- 실패를 200 응답으로 덮어서 조용히 삼키기
- 배포 후 회귀 없이 '올렸어요' 공지
- 팀장 승인 없이 main 병합 · VM 배포

맡는 파일 · 산출물

workflows/< 팀 >-handoff · < 팀 >-errors · 어댑터
① 설계서 5 · 7 절 (연동 · 오류)

81 · 82 일차 핵심 순간

81-4 · 81-7 VM 배포 1 · 2 회
81-6 장애 처리 (WF-04 · 09 · 10)

다른 역할에 넘겨줄 것

시험에게 : 배포 공지 (태그 · 바뀐 것)
팀장에게 : 배포 결과 · 남은 위험

시험 · 품질 담당 : 해야 할 것 · 하지 말아야 할 것

구현을 기다리는 사람이 아니라 , 처음부터 '된 것' 의 기준을 쥐고 있는 사람

해야 할 것

- 기대 결과를 먼저 쓰기 — cases/team.json 5 개 이상
- 지금 상태로 한 번 돌려 출발점 기록 (0 개여도 괜찮음)
- VM 배포마다 공통 + 팀 전체 회귀 → 통과 수 표
- 장애 시험 운전 — 켜고 , 돌리고 , 반드시 끄기
- LLM 시나리오 3 회 반복 · 흔들림 기록
- 실패는 6 칸 양식으로 담당자에게 , 고쳐지면 재시험해서 닫기

하지 말아야 할 것

- 구현이 끝나길 기다리며 손 놓기
- 결과에 맞춰 기대 결과를 고치기
- 3 번 중 2 번 통과를 '통과' 라고 쓰기
- 고장 스위치를 켜 둔 채 자리 비우기
- 기억 · 예상으로 결과서 숫자 채우기 (runs/ 에서만)

맡는 파일 · 산출물

cases/team.json · runs/ · requests/
③ 실행 결과서 · 데일리 '오늘의 숫자'

81 · 82 일차 핵심 순간

81-7 배포 직후 전체 회귀
82-3 · 4 시연 리허설 대본 진행

다른 역할에 넘겨줄 것

모두에게 : 실패 리포트 (재현 정보)
팀장에게 : 통과 수 · 흔들린 시나리오

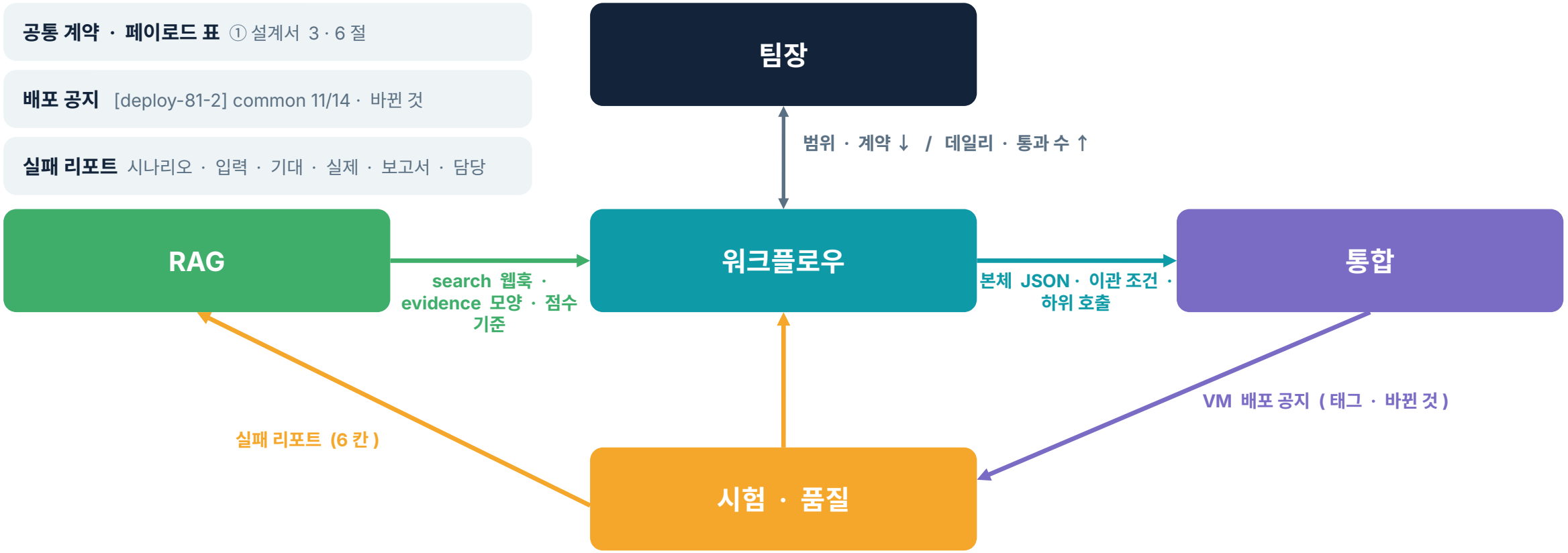
역할 사이에 주고받는 것

말로 넘기지 말고 , 정해진 모양 (파일 · 표 · 양식) 으로 넘깁니다

공통 계약 · 페이로드 표 ① 설계서 3 · 6 절

배포 공지 [deploy-81-2] common 11/14 · 바뀐 것

실패 리포트 시나리오 · 입력 · 기대 · 실제 · 보고서 · 담당



워크플로우 담당이 2 명 이상일 때 : 파일로 나눕니다

같은 JSON 을 둘이 고치면 n8n import 때 한쪽 작업이 사라집니다

나누기 A · 앞뒤로

<팀>-support — 1 번 담당

본체 : 입력 검사 · 분류 · 분기 · 응답

<팀>-answer — 2 번 담당

답변 · 검증 Agent (본체가 호출)

경계 : 분류 결과 + evidence → answer

나누기 B · 정상 / 예외

<팀>-support — 1 번 담당

정상 흐름 : 분류 → 검색 → 답변 → 검증

<팀>-guard — 2 번 담당

입력 검증 · 주입 차단 · 보완 질문 루프

경계 : guard 통과 여부 + 정리된 질문

1. 파일 경계 = 연결선 페이로드 표의 한 줄 (모양을 먼저 합의)
2. 본체가 하위를 부를 때는 timeout 과 오류 출력 (하위가 멈춰도 정해진 답)
3. 매일 1 교시에 경계의 모양이 바뀌었는지 서로 확인

역할과 상관없이 모두 지키는 규칙 7 가지

팀 헌장 4 절 ‘규칙’에 그대로 옮겨 적어도 됩니다



한 사람 = 한 파일

남의 파일은 말하고 고친다



공통 계약은 팀장 허락

바꿀 땐 설계서부터



고칠 때마다 공통 시나리오

커밋 메시지에 통과 수



키는 Credential 에만

커밋 전 tscan staged



30 분 막히면 팀 , 1 시간이면 강사

혼자 붙잡고 있지 않기



하루 끝 데일리 한 줄

한 일 · 숫자 · 막힌 것 · 내일



81 일차 8 교시 기능 동결

82 일차는 시연 · 문서 · 발표만

규칙을 어기면 고치는 사람은 ‘어긴 사람’ — 깨뜨린 사람이 고칩니다

최종 발표 20 분 , 역할별로 나눠 말합니다

모두가 자기 산출물을 설명할 수 있어야 합니다 — 82 일차 8 교시 평가 시험 준비도 됩니다



팀장	업무 시나리오 · 범위 · 조합 · 역할	2 분
워크플로우	흐름도 · 에이전트 역할 · 분기 · 병렬 · 루프 · 페이로드 한 줄	5 분
RAG	청킹 비교 숫자 · 임베딩 · 권한 필터 · 재색인	4 분
통합	외부 연동 · 오류 처리 · VM 배포 구성	3 분
시험 · 품질	시연 (정상 → 보완 질문 → 장애 → 이관) · 결과 숫자 · 실패 사례	6 분

+ 질의 5 분 — 질문은 그 부분을 맡은 사람이 답합니다