

문제를 보는 눈

워크샵 가이드북

박정수 (Park Jeongsoo) <toracle@gmail.com>

목차

1. 솔루션 중독 — Problem Space를 건너뛰는 습관	3
1.1. 3. 고정 케이스 실습	3
1.1.1. 워크시트 A-1 팀오더 케이스: 내 머릿속 지도 그리기	3
1.2. 4. 현업 케이스 슬롯	4
1.2.1. 워크시트 B-1 내 현업 케이스 등록	4
1.3. 6. 과제 안내	5
2. 모호함의 해부학 — 요구사항을 분해하는 언어	7
2.1. 워크시트 A-2 팀오더 케이스: 모호함 해부	7
2.2. 4. 현업 케이스 슬롯 (20분)	9
2.2.1. 워크시트 B-2 내 현업 케이스: FAC 분해	9
2.3. 6. 과제 안내 (5분)	10
3. 질문하는 자가 설계한다 — Context-Free Questions	11
3.1. 워크시트 A-3 팀오더 케이스: 인터뷰 + Black Box	11
3.2. 4. 현업 케이스 슬롯 (20분)	13
3.2.1. 워크시트 B-3 내 현업 케이스: Context-Free Questions 적용	14
3.3. 6. 과제 안내 (5분)	15
4. 지도를 그리기 전에 — 문제의 구조와 Domain Model	16
4.1. 워크시트 A-4 팀오더 케이스: 레이어 분석 + Domain Model	16
4.2. 4. 현업 케이스 슬롯 (20분)	18
4.2.1. 워크시트 B-4 내 현업 케이스: 레이어 분석 + Domain Model	18
4.3. 6. 과제 안내 (5분)	19
5. 안개 속에서 걷는 법 — 모호함을 안고 가기	21
5.1. 워크시트 A-5 팀오더 케이스: 모호함 진단과 Spike 설계	21
5.2. 4. 현업 케이스 슬롯 (20분)	23
5.2.1. 워크시트 B-5 내 현업 케이스: 모호함 재분류	23
5.3. 6. 과제 안내 (5분)	24
6. 이 문제, 어디서 본 것 같은데 — 유사 추론	25
6.1. 워크시트 A-6 팀오더 케이스: 유사 추론 + 문제 유형 매핑	25
6.2. 4. 현업 케이스 슬롯 (20분)	27
6.2.1. 워크시트 B-6 내 현업 케이스: 유사 추론 적용	27
6.3. 6. 과제 안내 (5분)	28
7. 어디에 공을 들여야 하는가 — Risk-Driven 설계	29
7.1. 워크시트 A-7 팀오더 케이스: Risk 식별 + 기법 매핑	29
7.2. 4. 현업 케이스 슬롯 (20분)	32
7.2.1. 워크시트 B-7 내 현업 케이스: Risk 식별 + 설계 에너지 배분	32
7.3. 6. 과제 안내 (5분)	32
8. 승현의 지도 — 해결책의 검증과 통합	34
8.1. 워크시트 A-8 팀오더 케이스: Problem-Solution Fit + Pre-mortem	34
8.2. 4. 현업 케이스 슬롯 — 최종 통합 (25분)	36
8.2.1. 워크시트 B-8 내 현업 케이스: 최종 통합 캔버스	36
8.3. 6. 워크샵 마무리 (5분)	37
9. 부록: 나의 현업 케이스☒—☒8강 누적 기록	39
9.1. 1강 이후☒—☒발견 메모	39
9.2. 2강 이후☒—☒발견 메모	39
9.3. 3강 이후☒—☒발견 메모	39

9.4. 4강 이후	발견 메모	39
9.5. 5강 이후	발견 메모	39
9.6. 6강 이후	발견 메모	40
9.7. 7강 이후	발견 메모	40
9.8. 8강 이후	발견 메모	40

대상: 풀스택 엔지니어 3-4년차 | 구성: 8강 × 100분

이 가이드북의 구성

이 가이드북은 워크샵 참가자용 자료입니다. 각 강의에서 직접 쓰고, 채우고, 기록하는 공간입니다.

워크시트 A 각 강마다 제공되는 고정 케이스 실습 양식
팀오더(TeamOrder) 케이스를 강마다 다른 도구로 탐색합니다

워크시트 B 1강부터 8강까지 동일한 현업 케이스를 들고 다니는 양식
매 강마다 새 도구를 내 현업 문제에 적용합니다
8강 끝에 하나의 완성된 분석이 남습니다

과제 카드 각 강 후 현업에서 해올 짧은 실천 과제

팀오더 케이스 소개

이 워크샵 전체를 관통하는 고정 케이스입니다.

회사: 기업용 SaaS 스타트업, 시리즈 A 직후

팀: 풀스택 엔지니어 4명, BizDev 1명

민준 (BizDev, 31세) 고객사와 계약을 따오는 사람
지연 (풀스택, 28세, 3년차) 빠르게 만드는 것에 자신 있다
승현 (풀스택, 32세, 5년차) 팀의 시니어 / 이 워크샵의 시점 인물
현태 (풀스택, 26세, 1년차) 막내. 당연한 것을 묻는다

1강 출발 요구사항:

민준: "팀 협업 기능 강화해주세요.
알림이랑 업무 배정 쪽이요. 다음 달 데모."

차례

강	제목	핵심 개념
1강	솔루션 중독 – Problem Space를 건너뛰는 습관	Problem Space vs Solution Space · Einstellung Effect · Presumptive Architecture

강	제목	핵심 개념
2강	모호함의 해부학 — 요구사항을 분해하는 언어	Functions / Attributes / Constraints (FAC) · 모호함의 세 출처
3강	질문하는 자가 설계한다 — Context-Free Questions	Context-Free Questions · Black Box Test · Jobs-to-be-Done
4강	지도를 그리기 전에 — 문제의 구조와 Domain Model	증상-문제-근본원인 레이어 · Problem Framing · Domain Model
5강	안개 속에서 걷는 법 — 모호함을 안고 가기	Resolvable vs Essential Ambiguity · Spike Solution · Technical Debt
6강	이 문제, 어디서 본 것 같은데 — 유사 추론	유사 추론(Analogical Reasoning) · 문제 유형 환원 · Trade-off 리서치
7강	어디에 공을 들여야 하는가 — Risk-Driven 설계	Quality Attributes · Engineering Risk · Risk-Driven Model
8강	승현의 지도 — 해결책의 검증과 통합	Problem-Solution Fit · Pre-mortem · 통합 회고

Chapter 1. 솔루션 중독 — Problem Space를 건너뛰는 습관

핵심 질문: 나는 지금 문제를 이해한 건가, 패턴을 인식한 건가?

핵심 개념: Problem Space vs Solution Space · Einstellung Effect · Presumptive Architecture

1.1.3. 고정 케이스 실습

1.1.1. 워크시트 A-1 | 팀오더 케이스: 내 머릿속 지도 그리기

워크시트 A-1 | 1강 / 팀오더 케이스

이름: _____

날짜: _____

팀 협업 기능 강화해주세요. 알림이랑 업무 배정 쪽이요. 다음 달 데모.

— 민준

PART 1. 지금 내 머릿속에 있는 것 (3분, 검열 없이)

이 요구사항을 듣는 순간 떠오른 것을 모두 적으세요.

(기술, 구조, 일정 추정, 무엇이든)

PART 2. 분류 (5분)

위에 적은 것들을 두 칸으로 나눠보세요.

Problem Space에 속하는 것 (문제 이해에 관한 것)	Solution Space에 속하는 것 (해결책에 관한 것)

PART 3. 내 Presumptive Architecture (5분)

"나는 이것이 _____ 종류의 시스템이라고 전제했다."

그 전제의 근거는?

- 과거에 비슷한 것을 만들어봤다
- 팀에서 이런 경우 보통 이렇게 한다
- 이 기술을 잘 안다
- 그냥 자연스럽게 떠올랐다
- 기타: _____

이 전제가 틀렸다면 무엇이 달라지는가?

PART 4. Problem Space에 남아있는 질문들 (10분)

아직 모르는 것, 물어봐야 할 것을 최대한 많이 적으세요.

(좋은 질문/나쁜 질문 없습니다. 수량이 목표입니다.)

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- ...

총 몇 개? _____

1.2. 4. 현업 케이스 슬롯

1.2.1. 워크시트 B-1 | 내 현업 케이스 등록

워크시트 B | 나의 현업 케이스 (8강 연속 사용)

이름: _____

최근 3개월 안에 받은 요구사항이나 요청 중,
 아직 명확하지 않다고 느끼거나,
 만들고 나서 "이게 아니었네" 했던 것 하나를 골라주세요.

케이스 제목 (한 줄):

원래 요구사항 (최대한 원문 그대로):

1강을 적용한다면:

처음 들었을 때 떠올린 것 (Solution Space):

당시 내 Presumptive Architecture는?

지금 돌아보면 물었어야 했던 질문들:

- 1.
- 2.
- 3.

1.3. 6. 과제 안내

과제 카드 #1

과제 카드 #1

이번 주에 받는 요구사항이나 업무 지시 중 하나를 골라서:

1. 처음 들은 순간 떠오른 Solution Space를 메모해두세요.
2. 그 요구사항에서 아직 모르는 것을 질문 목록으로 적어보세요. (최소 5개)
3. 워크시트 B에 그 케이스를 추가하거나, 이미 등록된 케이스를 계속 써도 됩니다.

다음 강 주제 예고:

요구사항의 해부학 \boxtimes — \boxtimes Functions, Attributes, Constraints

(다음 강으로)

Chapter 2. 모호함의 해부학 — 요구사항을 분해하는 언어

핵심 질문: 요구사항이 명확하다는 것은 어떤 의미인가?

핵심 개념: Functions / Attributes / Constraints (FAC) · 모호함의 세 출처

2.1. 워크시트 A-2 | 팀오더 케이스: 모호함 해부

워크시트 A-2 | 2강 / 팀오더 케이스

이름: _____

날짜: _____

팀 협업 기능 강화해주세요. 알림이랑 업무 배정 쪽이요. 다음 달 데모.

— 민준

팀장이 업무를 배정하면 팀원들이 모른대. 슬랙으로 따로 연락해야 하는데 너무 번거롭다고.

— 민준

PART 1. 모호함 출처 찾기 (7분)

위 두 문장에서 모호한 부분을 모두 찾아 표시하고,
어떤 종류의 모호함인지 분류해보세요.

모호한 표현	종류 (언어적 / 해석자간 / 생략된맥락)
1.	
2.	
3.	
4.	
5.	
6.	

PART 2. FAC 분해 (10분)

민준의 말에서 읽을 수 있는 것과,
아직 정해지지 않은 것(?)을 구분해서 채워보세요.

	말해진 것 (확인됨)	? (아직 모름)
Functions (무엇을)		
Attributes (어떻게)		
Constraints (조건/한계)		

"?" 칸이 "확인됨" 칸보다 많은가요?

- 네 (정상입니다. 탐색이 더 필요합니다)
- 아니오 (혹시 너무 빨리 채웠을 수도 있습니다 —재검토)

PART 3. 가장 위험한 공백은? (5분)

"?" 칸 중에서, 이것을 모르고 개발을 시작했을 때 재작업이나 실패 가능성이 가장 높은 것을 골라주세요.

가장 위험한 공백:

이유:

이것을 해소하려면 누구에게, 어떤 질문을 해야 하는가?

소그룹 공유 (8분)

3-4인 그룹으로 PART 2의 결과를 비교합니다.

비교 포인트

"같은 칸에 다른 것을 적은 경우가 있는가?"
→ 그 차이 자체가 해석자 간 모호함의 실례입니다.

"누군가 Attribute를 Function으로 적은 경우가 있는가?"
→ 왜 그렇게 분류했는지 이야기해보게 합니다.

"Constraints 칸이 가장 비어있는 경향이 있는가?"
→ 대부분 그렇습니다. Constraint는 가장 캐내기 어렵습니다.

"왜 이걸 생각이 안 났을까?"를 메타적으로 물어보세요.

전체 공유 (7분)

각 그룹에서 "가장 위험한 공백"으로 고른 것을 칠판에 모읍니다. 서로 다른 것을 골랐다면 — 왜 다른지가 토론의 핵심입니다.

2.2. 4. 현업 케이스 슬롯 (20분)

2.2.1. 워크시트 B-2 | 내 현업 케이스: FAC 분해

워크시트 B-2 | 2강 / 나의 현업 케이스 (1강에서 이어서)

이름: _____

1강에서 등록한 현업 케이스:

(제목을 다시 적어주세요)

원래 요구사항을 FAC로 분해해보세요.

	말해진 것 (확인됨)	? (아직 모름)
Functions		
Attributes		
Constraints		

돌아보기:

이 케이스를 처음 받았을 때, 내가 놓쳤던 FAC 항목은?

그것을 모르고 개발을 진행했을 때 실제로 어떤 일이 생겼는가?

(또는 생길 뻔했는가?)

만약 그때 이 도구가 있었다면, 가장 먼저 물었을 질문은?

2.3. 6. 과제 안내 (5분)

과제 카드 #2

과제 카드 #2

이번 주에 받은 요구사항 하나를
FAC 테이블로 분해해보세요.

특히 이것을 의식하세요:

1. Attributes 칸 - "어떻게 해야 한다"는
기준이 명시되어 있는가?
없다면 어떤 질문으로 끌어낼 수 있는가?
2. Constraints 칸 - "건드릴 수 없는 것"이
말해졌는가?
말해지지 않은 전제가 있다고 느껴지는가?
3. 워크시트 B에 결과를 추가해주세요.
다음 강에서 이어서 사용합니다.

다음 강 주제 예고:

FAC를 채우기 위한 질문 - 어떻게 물어야 하는가?

(다음 강으로)

Chapter 3. 질문하는 자가 설계한다 — Context-Free Questions

핵심 질문: BizDev이 원하는 것과 사용자가 필요한 것은 다르다 — 어떻게 캐내는가?

핵심 개념: Context-Free Questions · Black Box Test · Jobs-to-be-Done

3.1. 워크시트 A-3 | 팀오더 케이스: 인터뷰 + Black Box

실습 설정

2인 1조. 역할을 나눕니다.

- **인터뷰어 (승현 역):** Context-Free Questions를 사용해 민준을 인터뷰
- **BizDev (민준 역):** 아래 민준 카드를 받아서 그 정보 안에서만 대답

민준 역할 카드 (인터뷰어에게 보여주지 않음)

당신은 고객사 A사(중견 제조업체)와 상담하고 왔습니다. 아는 것:

- 고객사 팀장 3명이 각자 팀원 5-10명에게 업무를 배정함
- 현재는 시스템에서 배정 후, 슬랙으로 따로 연락함
- 팀원들이 업무를 놓치는 일이 주 1-2회 생김
- 팀장들이 현장(공장)에 자주 나감 → 모바일 사용
- 데모는 팀장들에게 보여줄 것 (팀원은 나중에)



모르는 것 (질문받으면 "잘 모르겠다"고 대답):

- 알림 응답 속도 기준
- 기존 시스템의 기술 스펙
- 예산
- 팀원들의 디바이스 환경

워크시트 A-3 | 3강 / 팀오더 케이스

이름 (인터뷰어): _____

이름 (BizDev): _____

PART 1. 인터뷰 (15분)

Context-Free Questions를 사용해 민준을 인터뷰하세요.

아래 세 카테고리에서 각 2개 이상씩 질문을 만들어, 실제로 사용해 보세요.

[Process Questions — 어떻게 일어나고 있는가]

질문	발견한 것
Q: A:	
Q: A:	
Q: A:	

[Product Questions — 무엇이어야 하는가]

질문	발견한 것
Q: A:	
Q: A:	
Q: A:	

[Meta-Questions — 질문에 대한 질문]

(최소 1개는 써보세요)

질문	발견한 것
Q: A:	

PART 2. Black Box 작성 (10분)

인터뷰에서 발견한 것을 바탕으로 채워보세요.

아직 모르는 것은 [?]로 표시.

기능 이름

Trigger (언제 시작되는가)

Input (무엇이, 누가 넣는가)

Output (무엇이, 누구에게 가는가)

예외 케이스 (동작 안 하는 조건)

1. _____
2. _____
3. _____

PART 3. 인터뷰 후 FAC 업데이트 (5분)

2강 워크시트와 비교해보세요.

인터뷰로 새로 채워진 것	여전히 [?]로 남은 것

이 인터뷰에서 가장 예상 밖의 발견은?

인터뷰 중 막힌 질문이 있었는가? 어떤 종류였는가?

소그룹 공유 (5분)

같은 역할 카드를 쓴 팀들이 모여 Black Box를 비교합니다.

비교 포인트

"같은 인터뷰 대상인데 Black Box가 얼마나 다른가?"

→ 질문의 방향이 달랐기 때문입니다. 어떤 질문이 어떤 발견을 만들었는지 역추적해보세요.

"예외 케이스를 가장 많이 찾은 팀은 어떤 질문을 했는가?"

→ "절대 해서는 안 되는 것"과 "이 기능이 실패하는 조건"을 물어본 팀이 예외를 더 많이 발견합니다.

"Meta-Question을 쓴 팀에서 어떤 일이 생겼는가?"

→ '제가 놓친 게 있나요?'가 가장 새로운 정보를 끌어냅니다. 이 경험을 전체 공유에서 꺼내세요.

3.2. 4. 현업 케이스 슬롯 (20분)

3.2.1. 워크시트 B-3 | 내 현업 케이스: Context-Free Questions 적용

워크시트 B-3 | 3강 / 나의 현업 케이스 (2강에서 이어서)

이름: _____

내 현업 케이스의 FAC에서 가장 중요한 [?] 항목:
(2강 워크시트에서 가져옴)

이 [?]를 채우기 위한 Context-Free Questions를 만들어보세요.

Process Questions (현재 어떻게 일어나고 있는가):

1. Q1. _____

2. Q2. _____

Product Questions (무엇이어야 하는가):

1. Q1. _____

2. Q2. _____

Meta-Question (내가 놓치고 있는 것):

1. Q1. _____

이 질문들을 실제로 할 수 있는 사람은 누구인가?

대상	언제/어떻게 물어볼 것인가

내 케이스의 Black Box를 그려보세요.
(현재 파악된 정보 기준. [?] 적극 사용)

기능 이름

Trigger

Input

Output

예외 케이스

1. _____
 2. _____
-

3.3. 6. 과제 안내 (5분)

과제 카드 #3

과제 카드 #3

이번 주에 BizDev, 기획자, 또는 팀원 중 한 명과 짧은 대화를 해보세요.

대화 대상:

내 현업 케이스의 [?] 중 하나를 해소할 수 있는 사람 (워크시트 B-3 참고)

시도할 것:

1. Context-Free Questions 중 하나를 자연스럽게 사용해보기
2. 대화 후 발견한 것을 워크시트 B에 추가 — 특히 예상 밖의 답변이 있었다면 기록
3. 가능하면 Black Box를 업데이트해오기

다음 강 주제 예고:

문제의 구조 파악 — 증상, 문제, 근본원인, 그리고 Domain Model

(다음 강으로)

Chapter 4. 지도를 그리기 전에 — 문제의 구조와 Domain Model

핵심 질문: 문제를 충분히 이해했는지 어떻게 아는가?

핵심 개념: 증상-문제-근본원인 레이어 · Problem Framing · Domain Model

4.1. 워크시트 A-4 | 팀오더 케이스: 레이어 분석 + Domain Model

워크시트 A-4 | 4강 / 팀오더 케이스

이름: _____

날짜: _____

지금까지 파악한 것 (1-3강 요약):

- 팀장이 시스템에서 업무 배정 후 슬랙으로 따로 연락
- 팀원들이 현장에서 태블릿/폰 사용
- 업무 놓치면 라인 스탑 발생
- 야간 긴급 배정 케이스 있음
- 다음 달 팀장들에게 데모 예정

PART 1. 레이어 분류 (8분)

아래 현상들을 증상 / 문제 / 근본원인으로 분류하세요.
(정답이 하나가 아닐 수 있습니다 — 팀과 토론하세요)

현상	레이어	이유
팀원이 업무를 놓친다		
배정 후 슬랙으로 따로 연락한다		
라인 스탑이 발생한다		
팀원들이 현장에서 슬랙을 잘 안 본다		
배정 흐름이 시스템 밖에서 일어난다		

이 현상들 중, 우리가 해결해야 할 레이어는 어디인가?
(데모 목적, 1개월 일정을 고려해서)

선택한 레이어

이유

PART 2. Problem Framing (7분)

"팀원들이 업무를 놓친다"는 현상을 다른 프레임으로 볼 수 있는 방법을 2가지 이상 써보세요.

프레임 1

이 프레임에서 나오는 해결책

프레임 2

이 프레임에서 나오는 해결책

프레임 3 (있다면)

이 프레임에서 나오는 해결책

팀이 선택한 프레임은? 왜?

PART 3. Domain Model 스케치 (15분)

팀오더 케이스의 Domain Model을 그려보세요.

아래 카테고리를 채우되, 네모칸 안에 자유롭게 그려도 됩니다.

Actor (누가 관여하는가)

Action (어떤 행위가 일어나는가)

State (업무가 가질 수 있는 상태들)

Rule (지켜져야 하는 규칙)

Object (시스템이 다루는 주요 개념/항목)

체크: 팀원 모두가 같은 어휘를 쓰고 있는가?

헛갈리거나 팀마다 다르게 부르는 개념이 있었는가?

그 개념을 어떻게 정의하기로 했는가?

소그룹 공유 (7분)

각 그룹의 Domain Model을 칠판에 나란히 붙이거나 그립니다.

비교 포인트

"상태(State)를 얼마나 다르게 그렸는가?"

→ State 설계가 가장 많이 같립니다. '배정됨'과 '확인됨'을 같은 것으로 본 팀 vs 다른 것으로 본 팀. 이 차이가 나중에 DB 설계, API 설계를 완전히 바꿉니다.

"Rule을 얼마나 찾았는가?"

→ 야간 긴급 규칙을 넣은 팀이 있는가? 없는 팀은 - 왜 빠졌는지 돌아보게 합니다.

"지금 그린 Domain Model이 현재 설계(있다면)와 얼마나 다른가?"

→ 괴리가 큰 것이 나중에 재작업 포인트입니다.

4.2. 4. 현업 케이스 슬롯 (20분)

4.2.1. 워크시트 B-4 | 내 현업 케이스: 레이어 분석 + Domain Model

워크시트 B-4 | 4강 / 나의 현업 케이스 (3강에서 이어서)

이름: _____

내 케이스에서 관찰되는 현상들을 레이어로 분류해보세요.

관찰된 현상	증상/문제/근본원인	확신도(상/중/하)
1.		
2.		
3.		

관찰된 현상	증상/문제/근본원인	확신도(상/중/하)
4.		

우리가 실제로 해결한 레이어는?

- 증상 레이어
- 문제 레이어
- 근본원인 레이어

그 선택이 의식적이었는가, 아니면 나중에 보니 그랬는가?

내 케이스의 Domain Model을 스케치해보세요.

Actor

Action

State

Rule

Object

돌아보기:

지금 그린 Domain Model과 실제 설계/구현의 괴리는?

그 괴리가 나중에 어떤 문제를 만들었는가?

(또는 만들 가능성이 있는가?)

4.3. 6. 과제 안내 (5분)

과제 카드 #4

과제 카드 #4

내 현업 케이스의 Domain Model을 다듬어오세요.

특히 이것을 확인하세요:

1. State — 내가 다루는 핵심 개념(예: 업무, 주문, 사용자)이 가질 수 있는 상태들을 빠짐없이 나열했는가?
2. Rule — 현실에서 지켜지는 규칙 중 시스템에 반영되지 않은 것이 있는가?
3. 팀원 한 명에게 내 Domain Model을 보여주고 "이게 맞아요?"라고 물어보세요. 다르게 이해한 부분이 있다면 기록해오세요.

다음 강 주제 예고:

모호함을 안고 가기 — 해소할 수 없는 모호함은 어떻게 다루는가? (Cunningham)

(다음 강으로)

Chapter 5. 안개 속에서 걷는 법 — 모호함을 안고 가기

핵심 질문: 해소할 수 없는 모호함은 어떻게 다루는가?

핵심 개념: Resolvable vs Essential Ambiguity · Spike Solution · Technical Debt

5.1. 워크시트 A-5 | 팀오더 케이스: 모호함 진단과 Spike 설계

워크시트 A-5 | 5강 / 팀오더 케이스

이름: _____

날짜: _____

팀오더 케이스의 현재 상태 (1-4강 누적):

- Domain Model 대략 그려짐
- Black Box 초안 있음
- FAC 중 Attributes 일부 아직 [?]
- 다음 달 데모 목표

PART 1. 모호함 목록 작성 (5분)

지금 이 케이스에서 아직 해소되지 않은 모호함들을 생각나는 대로 적어보세요.

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____

PART 2. 진단 — Resolvable vs Essential (8분)

각 모호함에 대해 진단 질문을 적용하세요.

진단 질문:

"이것을 알기 위해 내가 할 수 있는 행동이 있는가?"

모호함	종류	근거
1.	* [] R * [] E	
2.	* [] R * [] E	
3.	* [] R * [] E	

모호함	종류	근거
4.	* <input type="checkbox"/> R * <input type="checkbox"/> E	
5.	* <input type="checkbox"/> R * <input type="checkbox"/> E	
6.	* <input type="checkbox"/> R * <input type="checkbox"/> E	

R = Resolvable (물어보면/조사하면 해소 가능)

E = Essential (해봐야/만들어봐야 안다)

PART 3. Spike 설계 (12분)

Essential로 분류된 것 중 가장 중요한 것 하나를 골라 Spike를 설계해보세요.

탐색할 모호함:

가장 단순한 탐색 행동은 무엇인가?
(버릴 것을 알고, 최대한 빠르게, 최소한으로)

이 Spike가 성공했다면 어떤 상태인가?
(무엇을 알게 되면 "됐다"고 할 수 있는가)

이 Spike가 실패했다면 어떤 상태인가?
(무엇이 안 됐을 때 "아니다"라고 할 수 있는가)

예상 소요 시간: * 1-2시간 * 반나절 * 하루 * 그 이상

Spike 결과로 다음에 무엇을 결정할 수 있는가?

PART 4. "지금 이 결정이 꼭 필요한가?" 체크 (5분)

Essential로 분류된 나머지 모호함들 중, 지금 당장 결정하지 않아도 되는 것들은?

(데모까지 1개월. 이 모호함이 데모에 영향을 주는가?)

모호함	지금 결정 필요?	이유
	* <input type="checkbox"/> 예 * <input type="checkbox"/> 아니오	
	* <input type="checkbox"/> 예 * <input type="checkbox"/> 아니오	
	* <input type="checkbox"/> 예 * <input type="checkbox"/> 아니오	

소그룹 공유 (5분)

비교 포인트

"같은 모호함을 R로 분류한 팀과 E로 분류한 팀이 있는가?"

→ 왜 다르게 봤는지가 핵심입니다.

실제로 행동이 가능한데 E로 분류한 것은 없는가?

반대로, 아무도 모르는데 R로 분류한 것은 없는가?

"Spike를 설계할 때 가장 막힌 것은 무엇인가?"

→ '가장 단순한'의 기준을 잡는 것.

"이것보다 더 단순하게 할 수 있는가?"를

반복해서 묻도록 유도합니다.

"'지금 결정 안 해도 된다'는 칸을 활용했는가?"

→ 이 판단 자체가 중요한 역량입니다.

모든 모호함을 지금 해소해야 한다는 강박에서 벗어나는 것.

5.2.4. 현업 케이스 슬롯 (20분)

5.2.1. 워크시트 B-5 | 내 현업 케이스: 모호함 재분류

워크시트 B-5 | 5강 / 나의 현업 케이스 (4강에서 이어서)

이름: _____

내 케이스의 미해소 모호함 목록

(2-4강 워크시트의 [?]들을 여기로 옮겨주세요):

1. _____
2. _____
3. _____
4. _____
5. _____

각 모호함을 진단하세요.

모호함	R/E	진단 근거
1.	* [] R * [] E	
2.	* [] R * [] E	
3.	* [] R * [] E	
4.	* [] R * [] E	
5.	* [] R * [] E	

Essential로 분류된 것 중 하나를 골라:

그 당시 이것을 어떻게 다뤘는가?

- 그냥 가정하고 넘어갔다
- 계속 물어봤는데 답이 안 나왔다
- 일단 만들어서 확인했다
- 결정을 미뤘다
- 기타: _____

지금 돌아보면, 가장 좋은 접근은 어떤 것이었을까?

그 Essential Ambiguity에 대한 Spike를 설계한다면?

가장 단순한 탐색 행동:

5.3. 6. 과제 안내 (5분)

과제 카드 #5

과제 카드 #5

이것은 회고 과제입니다.

최근 6개월 안에 진행한 작업 중, "해봐야 알았던" 경험을 하나 찾아보세요.

적어올 것:

1. 그 모호함은 Resolvable이었는가, Essential이었는가?
2. 당시 어떻게 다뤘는가? (가정/질문/그냥 만들기/미루기 중)
3. Spike를 설계했다면 어떻게 했을까? "가장 단순한 탐색 행동"을 지금 설계해보세요.

다음 강 주제 예고:

이 문제, 어디서 본 것 같은데 — 유사 추론과 비즈니스 리서치

(다음 강으로)

Chapter 6. 이 문제, 어디서 본 것 같은데 — 유사 추론

핵심 질문: 이 문제를 처음 보는 게 아닐 수 있다 — 누가 이미 풀었는가?

핵심 개념: 유사 추론(Analogical Reasoning) · 문제 유형 환원 · Trade-off 리서치

6.1. 워크시트 A-6 | 팀오더 케이스: 유사 추론 + 문제 유형 매핑

워크시트 A-6 | 6강 / 팀오더 케이스

이름: _____

날짜: _____

팀오더의 핵심 문제 구조 (복습):

"현장 근무자에게 긴급도가 다른 업무 알림을 불확실한 네트워크/절전 환경에서 확실하게 전달하고, 수신 여부를 팀장이 확인할 수 있어야 한다"

PART 1. 유사 추론 (15분)

위 구조와 닮은 문제를 다른 도메인에서 찾아보세요. 각 유사 사례에서 배울 수 있는 원리를 추출하세요.

유사 사례 1:

도메인: _____

어떤 점이 닮았는가: _____

그 사례에서 쓴 해결 원리: _____

팀오더에 역수입할 수 있는 것: _____

유사 사례 2:

도메인: _____

어떤 점이 닮았는가: _____

그 사례에서 쓴 해결 원리: _____

팀오더에 역수입할 수 있는 것: _____

유사 사례 3:

도메인: _____

어떤 점이 닮았는가: _____

그 사례에서 쓴 해결 원리: _____

팀오더에 역수입할 수 있는 것: _____

PART 2. 문제 유형 매핑 (5분)

팀오더의 핵심 문제를 문제 유형으로 환원해보세요. (복수 선택 가능)

Matching

- Trust
- Cold Start
- Visibility
- Rate Control
- Consistency
- 기타: _____

선택한 유형에서 알려진 해결 패턴은?

PART 3. 비즈니스 리서치 (10분)

유사한 알림/업무 배정 문제를 다루는 서비스를 하나 골라서 Trade-off 관점으로 분석해보세요. (예: 슬랙, Asana, PagerDuty, 네이버 워크스, 카카오워크 등)

분석 대상: +++ _____ +++

이 서비스가 최우선으로 한 것: +
+++ _____ +++

포기한 것 (또는 약한 부분): +
+++ _____ +++

최적화된 사용자/환경: +
+++ _____ +++

팀오더의 사용자(공장 현장 근무자)에게
이 서비스 설계가 맞지 않는 이유: +
+++ _____ +++

팀오더가 이 서비스와 달라야 하는 한 가지: +
+++ _____ +++

소그룹 공유 (10분)

비교 포인트

"팀마다 다른 유사 사례를 찾았는가?"

- ↳ 같은 문제에서 전혀 다른 도메인의 유사 사례가 나올 수 있습니다.
- 병원 트리아지를 찾은 팀과 TCP 재전송을 찾은 팀이 있다면 둘 다 맞습니다. 그리고 둘에서 끌어낸 원리가 다를 것입니다.
- 그 차이를 칠판에 나열해보세요.

"원리를 역수입할 때 막힌 부분은?"

→ "해결책을 복사하려 했는가, 원리를 가져오려 했는가?"를
확인합니다. 전자는 대부분 "우리 상황에 안 맞아요"로 끝납니다.

"비즈니스 리서치에서 Trade-off를 읽는 것이
단순한 기능 비교와 어떻게 달랐는가?"

→ 이 질문을 꺼내서 리서치 관점의 차이를 언어화하게 합니다.

6.2. 4. 현업 케이스 슬롯 (20분)

6.2.1. 워크시트 B-6 | 내 현업 케이스: 유사 추론 적용

워크시트 B-6 | 6강 / 나의 현업 케이스 (5강에서 이어서)

이름: _____

내 케이스의 핵심 문제 구조를 한 문장으로:

(4강 Domain Model + 5강 모호함 진단을 바탕으로)

이 구조와 닮은 문제를 다른 도메인에서 2개 이상 찾아보세요.

유사 사례 1:

도메인 / 서비스: _____

닮은 구조: _____

역수입할 원리: _____

유사 사례 2:

도메인 / 서비스: _____

닮은 구조: _____

역수입할 원리: _____

내 케이스를 문제 유형으로 환원한다면?

문제 유형: +++ _____ +++

이 유형에서 자주 쓰이는 해결 패턴: +

+++ _____ +++

돌아보기:

이 케이스를 처음 받았을 때, 유사 추론을 했는가?

했다 — 어떤 유사 사례를 참고했는가? _____

안 했다 — 지금 찾은 유사 사례가 당시에 도움이 됐을까? _____

가장 유용했던 유사 사례와 그 이유:

6.3. 6. 과제 안내 (5분)

과제 카드 #6

과제 카드 #6

내 현업 케이스에 대해:

1. 찾아낸 유사 사례 중 가장 유용한 것의 원리를 내 케이스에 실제로 적용해보세요. 워크시트 B에 "역수입한 원리 적용 결과"를 추가해주세요.
2. (선택) 비슷한 문제를 다루는 서비스나 오픈소스를 하나 찾아보세요. 기능 목록이 아니라 Trade-off 관점으로 2-3문장으로 정리해주세요.

다음 강 주제 예고:

얼마나 깊이 설계할 것인가 — Risk-Driven으로 Solution Space 탐색하기

(다음 강으로)

Chapter 7. 어디에 공을 들여야 하는가 — Risk-Driven 설계

핵심 질문: 모든 요구사항을 같은 깊이로 설계해야 하는가?

핵심 개념: Quality Attributes · Engineering Risk · Risk-Driven Model

7.1. 워크시트 A-7 | 팀오더 케이스: Risk 식별 + 기법 매핑

워크시트 A-7 | 7강 / 팀오더 케이스

이름: _____

날짜: _____

팀오더의 현재 상황 요약 (1-6강 누적):

- 핵심 기능: 업무 배정 알림 + 수신 확인(acknowledgement)
- 사용자: 공장 현장 팀장(배정) / 팀원(수신, 태블릿/폰)
- 환경: 불안정한 와이파이, 절전 모드, 야간 운영 있음
- 알림 채널: 일반은 FCM 앱 푸시, 야간 긴급은 미결
- 일정: 1개월 후 데모

PART 1. 잠재적 실패 리스크 식별 (7분)

이 시스템이 실패할 수 있는 방법을 최대한 많이 적어보세요.

(좋은/나쁜 리스크 없음. 수량이 목표)

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.

PART 2. 리스크 분류 (5분)

위의 리스크를 분류하세요.

리스크	E / PM	Quality Attribute (해당 시)
1.	* <input type="checkbox"/> E * <input type="checkbox"/> PM	
2.	* <input type="checkbox"/> E * <input type="checkbox"/> PM	
3.	* <input type="checkbox"/> E * <input type="checkbox"/> PM	
4.	* <input type="checkbox"/> E * <input type="checkbox"/> PM	
5.	* <input type="checkbox"/> E * <input type="checkbox"/> PM	

리스크	E / PM	Quality Attribute (해당 시)
6.	* [] E * [] PM	
7.	* [] E * [] PM	
8.	* [] E * [] PM	

E = Engineering Risk PM = Project Management Risk

PART 3. 우선순위 + 기법 매핑 (10분)

Engineering Risk 중 상위 3개를 골라서, 각각에 맞는 탐색/검증 기법을 매핑하세요.

리스크 #1 (가장 중요):

이 리스크가 실현되면:

대응 기법:

- 프로토타입 / Spike
- 부하/성능 테스트
- 아키텍처 리뷰
- 보안 분석 (Threat model)
- 실제 환경 테스트
- 다이어그램/모델링
- 기타:

이 기법을 언제, 어떻게 할 것인가:

리스크 #2:

이 리스크가 실현되면:

대응 기법:

- 프로토타입 / Spike
- 부하/성능 테스트
- 아키텍처 리뷰
- 보안 분석 (Threat model)
- 실제 환경 테스트
- 다이어그램/모델링
- 기타:

이 기법을 언제, 어떻게 할 것인가:

리스크 #3:

이 리스크가 실현되면:

대응 기법:

- 프로토타입 / Spike
- 부하/성능 테스트
- 아키텍처 리뷰
- 보안 분석 (Threat model)
- 실제 환경 테스트
- 다이어그램/모델링
- 기타:

이 기법을 언제, 어떻게 할 것인가:

PART 4. "지금 설계하지 않아도 되는 것" (8분)

리스크 체크리스트 기준으로:

이번 데모(1개월)에서 리스크가 낮아서

지금 당장 설계 에너지를 쓰지 않아도 되는 것들은?

(예: "확장성 — 지금 사용자가 15명이므로 데모에서는 무관")

항목	이유
1.	
2.	
3.	

이것들을 언제 다시 검토할 것인가?

소그룹 공유 (8분)

비교 포인트 ————— "팀마다 상위 3개 리스크가 다른가?" → 같은 시스템인데 리스크 우선순위가 다르다면 어떤 맥락 차이가 그 판단을 만들었는가? 데모 관점 vs 실서비스 관점의 차이일 수 있습니다.

"PM Risk와 Engineering Risk를 혼동한 경우가 있는가?" → 가장 자주 섞이는 것: "1개월 안에 못 만들 것 같다" → PM Risk "야간 알림 전달을 보장 못 할 것 같다" → Engineering Risk 이 둘에 대응하는 기법이 완전히 다름을 짚어줍니다.

"PART 4 — '지금 안 해도 된다'는 결정이 어려웠는가?" → 어려웠다면 — 왜 어려웠는가? "완벽하게 설계해야 한다"는 압박과 어떻게 타협하는가? 이것이 이 강의 핵심 인식 전환입니다.

7.2. 4. 현업 케이스 슬롯 (20분)

7.2.1. 워크시트 B-7 | 내 현업 케이스: Risk 식별 + 설계 에너지 배분

워크시트 B-7 | 7강 / 나의 현업 케이스 (6강에서 이어서)

이름: _____

내 케이스의 Engineering Risk 3가지:

1. Quality Attribute: 대응 기법:
2. Quality Attribute: 대응 기법:
3. Quality Attribute: 대응 기법:

돌아보기:

이 케이스를 진행할 때, 실제로 설계 에너지를 어디에 썼는가?

가장 많이 설계한 부분:

가장 빨리 넘어간 부분:

Risk-Driven 관점으로 보면 —역전이 필요한 부분이 있는가?

(많이 설계한 것 중 리스크가 낮은 것 /

빨리 넘어간 것 중 리스크가 높은 것)

7.3. 6. 과제 안내 (5분)

과제 카드 #7

과제 카드 #7

내 현업 케이스의 Engineering Risk 중
가장 중요한 것 하나를 골라서:

1. 이 리스크에 대응하는 기법을 실제로 적용해보세요.
(Spike, 프로토타입, 다이어그램 작성 등)

2. 그 결과로 리스크가 낮아졌는가, 아니면 새로운 리스크가 발견됐는가를 기록해오세요.

3. 워크시트 B에 "리스크 대응 결과"를 추가하세요.

다음 강 주제 예고:

승현의 지도 -해결책을 원래의 문제에 대입하고
전체를 통합하는 법 (마지막 강)

(다음 강으로)

Chapter 8. 승현의 지도 — 해결책의 검증과 통합

핵심 질문: 해결책이 그럴듯해 보이는 것과 실제로 작동하는 것은 다르다

핵심 개념: Problem-Solution Fit · Pre-mortem · 통합 회고

8.1. 워크시트 A-8 | 팀오더 케이스: Problem-Solution Fit + Pre-mortem

워크시트 A-8 | 8강 / 팀오더 케이스

이름: _____

날짜: _____

팀오더 최종 해결책 (7강까지의 결론):

- 일반 업무 배정 → FCM 앱 푸시 (수신 확인 포함)
- 야간 긴급 → SMS fallback (미확인 시 자동 전환)
- 알림 채널 인터페이스 추상화
- 데모 범위: 위 세 가지
- 확장성 설계: 데모 후로 미룸

PART 1. Problem-Solution Fit 체크 (10분)

1-4강에서 정의한 문제와 지금의 해결책을 대조해보세요.

문제 정의에서	해결책에서 대응되는가?	어떻게 대응됐는가
팀원이 업무를 놓쳐 라인 스태프	* <input type="checkbox"/> 대응됨 * <input type="checkbox"/> 부분적 * <input type="checkbox"/> 없음	
배정 후 별도 연락 필요	* <input type="checkbox"/> 대응됨 * <input type="checkbox"/> 부분적 * <input type="checkbox"/> 없음	
야간 긴급 배정 케이스	* <input type="checkbox"/> 대응됨 * <input type="checkbox"/> 부분적 * <input type="checkbox"/> 없음	
팀장이 수신 여부를 모름	* <input type="checkbox"/> 대응됨 * <input type="checkbox"/> 부분적 * <input type="checkbox"/> 없음	
모바일/태블릿 환경	* <input type="checkbox"/> 대응됨 * <input type="checkbox"/> 부분적 * <input type="checkbox"/> 없음	

"만들지 않은 것" 중 나중에 다시 볼 것들:

(7강 PART 4에서 미룬 것들)

항목	예상 재검토 시점
1.	
2.	
3.	

PART 2. Pre-mortem (12분)

"팀오더 알림 기능을 데모에서 보여줬고,
실서비스에 반영했습니다.
3개월이 지났습니다. 실패했습니다."

이유가 무엇일까요? 최대한 많이 적어보세요. (3분 개인 작업)

1.
2.
3.
4.
5.

소그룹에서 가장 많이 나온 실패 이유 상위 3개:

1.
2.
3.

이 중 지금 미리 대응 가능한 것:

실패 이유	대응 방법

PART 3. 1강과 지금을 비교 (8분)

1강에서 "30초 안에 떠올린 것"을 기억하시나요?

(또는 1강 워크시트 A-1 Part 1을 꺼내보세요)

당시 30초 안에 떠올린 것들:

지금의 해결책과 비교:

같은 것:

달라진 것 (추가된 것, 빠진 것, 바뀐 것):

문제를 더 깊이 이해한 결과로:

- 해결책이 더 단순해졌다
- 해결책이 더 복잡해졌다
- 만들지 않아도 되는 것이 생겼다
- 처음에 생각했던 것 중 틀린 것이 있었다
- 기타:

이 워크샵에서 내가 가장 크게 바뀐 것 한 가지:

소그룹 공유 (8분)

이 강의 핵심 비교 포인트 ————— "PART 3에서 '만들지 않아도 되는 것이 생겼다'에 체크한 사람이 있는가?" → 이것이 이 강의, 그리고 워크샵 전체의 핵심 결론입니다. 그들에게 물어보세요: "무엇이 사라졌나요? 왜 사라져도 된다고 알게 됐나요?"

"해결책이 더 단순해졌는가?" → 더 단순해진 팀: 어느 강에서 그 단순화가 생겼는가? (2강 FAC? 4강 레이어 선택? 5강 Spike? 7강 Risk?)

"Pre-mortem에서 나온 실패 이유 중 2강-4강의 도구로 미리 잡을 수 있는 것이 있는가?" → 이 연결이 보이면 워크샵 전체의 학습이 통합된 것입니다.

8.2. 4. 현업 케이스 슬롯 — 최종 통합 (25분)

8강의 현업 케이스 슬롯은 평소와 다릅니다. 워크시트 B-1부터 B-7을 모두 꺼내서 하나의 케이스 완성분을 만드는 시간입니다.

8.2.1. 워크시트 B-8 | 내 현업 케이스: 최종 통합 캔버스

워크시트 B-8 | 8강 / 나의 현업 케이스 — 완성분

이름: _____

케이스 제목:

원래 요구사항 (1강에서):

문제 이해 (Problem Space 탐색 결과)

FAC 요약 (2강):

Functions:

Attributes (핵심):

Constraints (핵심):

해결 레이어 (4강):

* [] 증상 * [] 문제 * [] 근본원인 선택 이유:

Domain Model 핵심 요소 (4강):

핵심 상태(State):

핵심 규칙(Rule):

주요 모호함과 처리 (5강):

Resolvable로 해소한 것:

Essential로 Spike한 것:

미루기로 결정한 것:

문제 유형 환원 (6강):

해결책 (Solution Space 탐색 결과)

핵심 Engineering Risk 3개 (7강):

1.
2.
3.
지금 만들 것:

나중에 만들 것 + 재검토 시점:

만들지 않기로 한 것 + 이유:

Problem-Solution Fit 체크 (8강):

문제의 핵심이 해결책에 반영됐는가? * [] 예 * [] 부분적 * [] 아니오

만들지 않은 것 중 지금 당장 필요한 것이 빠졌는가?

회고:

이 케이스를 처음 받았을 때와 비교해서,
문제 이해가 해결책을 어떻게 바꿨는가?

단순해진 것:

사라진 것 (만들지 않아도 됨을 알게 된 것):

더 명확해진 것:

발표 (선택, 10분)

희망자가 자신의 케이스 완성본을 5분 내외로 발표합니다.

발표 구조 제안:

1. 원래 요구사항은 어떤 것이었나
2. 문제를 이해하면서 가장 크게 바뀐 것은
3. 결과적으로 만들기로 한 것과 만들지 않기로 한 것
4. 이 케이스에서 배운 한 가지

8.3. 6. 워크샵 마무리 (5분)

과제 카드 #8☒—☒이번엔 과제가 아닙니다

마지막 카드

과제가 없습니다.

대신, 이것 하나를 가지고 가세요.

다음 번에 요구사항을 받았을 때,
Solution Space로 넘어가기 전에
딱 한 번만 멈추고 물어보세요.

"나는 지금 무엇을 알고 있는가?
그리고 무엇을 아직 모르는가?"
그 질문이 이 워크샵 전체입니다.

Chapter 9. 부록: 나의 현업 케이스 - 8강 누적 기록

워크시트 B는 8강을 거치며 같은 케이스를 점진적으로 심화합니다. 이 페이지는 강별 핵심 발견을 한눈에 정리하는 공간입니다.

9.1. 1강 이후 - 발견 메모

적용 도구: Problem Space vs Solution Space · Einstellung Effect · Presumptive Architecture

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.2. 2강 이후 - 발견 메모

적용 도구: Functions / Attributes / Constraints (FAC) · 모호함의 세 출처

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.3. 3강 이후 - 발견 메모

적용 도구: Context-Free Questions · Black Box Test · Jobs-to-be-Done

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.4. 4강 이후 - 발견 메모

적용 도구: 증상-문제-근본원인 레이어 · Problem Framing · Domain Model

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.5. 5강 이후 - 발견 메모

적용 도구: Resolvable vs Essential Ambiguity · Spike Solution · Technical Debt

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.6. 6강 이후—발견 메모

적용 도구: 유사 추론(Analogical Reasoning) · 문제 유형 환원 · Trade-off 리서치

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.7. 7강 이후—발견 메모

적용 도구: Quality Attributes · Engineering Risk · Risk-Driven Model

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):

9.8. 8강 이후—발견 메모

적용 도구: Problem-Solution Fit · Pre-mortem · 통합 회고

발견한 것:

바뀐 것 (문제 이해, 해결책 방향):