

AI-Native 프로젝트 기획

AI 시대 살아남는 제품 만들기

카카오 신입 공채 AI PM 강의

강사: 쫘스 | Email: hosohere@gmail.com

Hi, I'm 쫘스



Session 1

PM 눈으로 세상 바라보기

강사: 쫘스 | Email: hosohere@gmail.com

모든 것은 기획이다.

MVP, 사업 개발, 마케팅, 콘텐츠, 인생까지...
기획은 세상을 다르게 바라보게 하는 프레임이다.

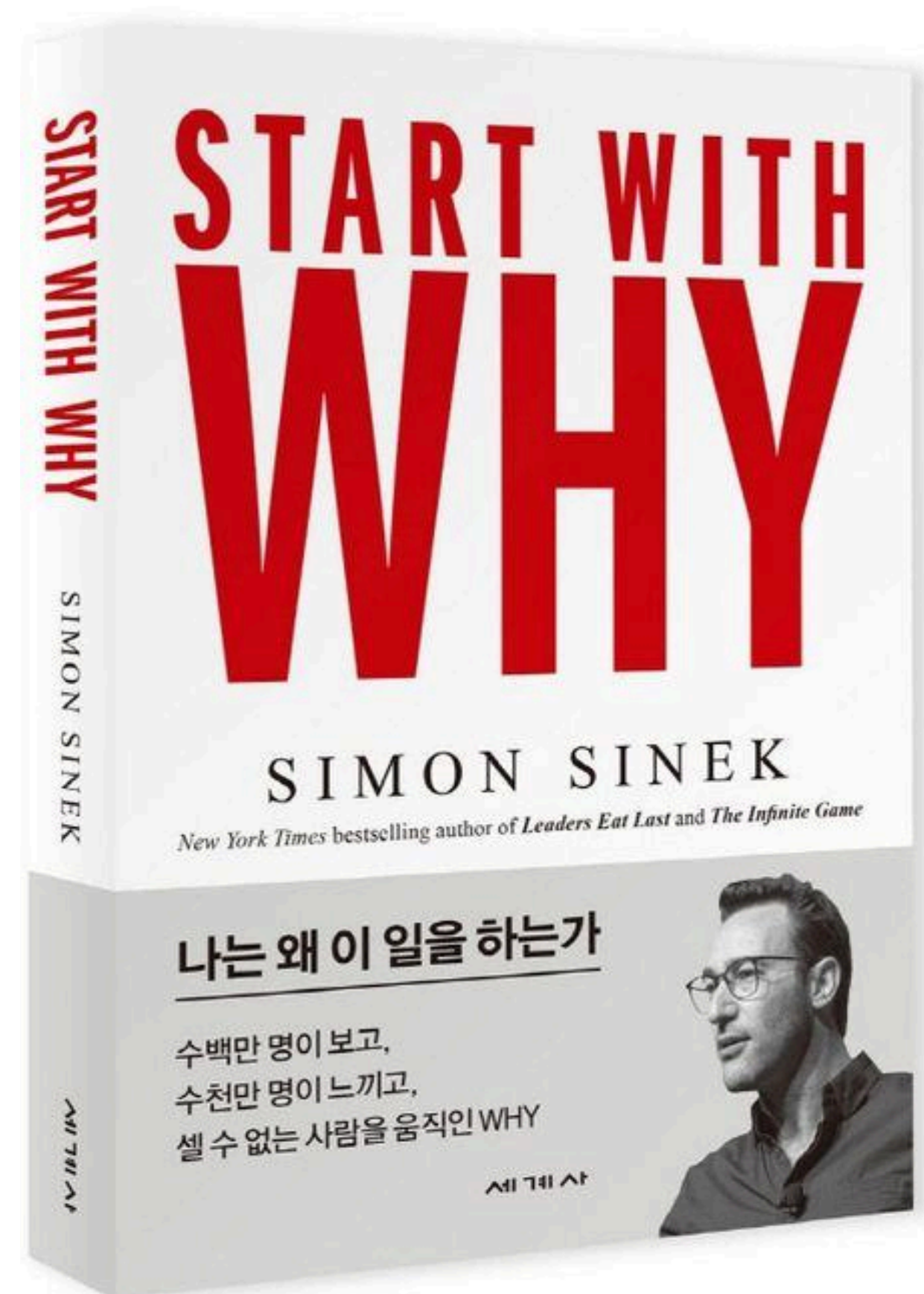
프레임(Frame)



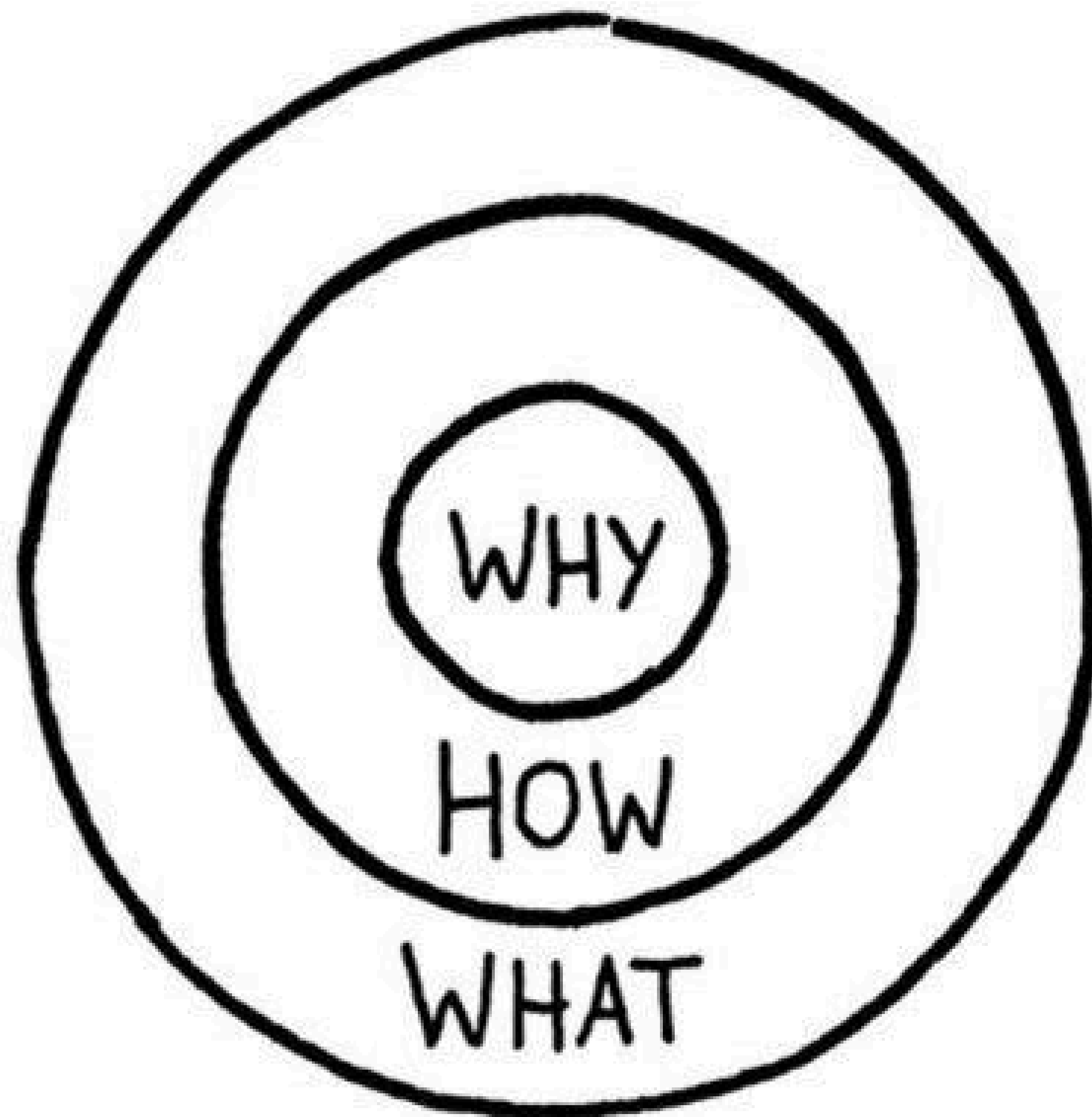
문제 정의는 상대적으로 어렵다

조직의 생산성을 높이기 위한 자동화를 하면서 가장 크게 깨달은 점은, **엔지니어는 단순히 주어진 문제를 해결하는 것을 넘어, 진짜 문제를 발굴하고 정확히 정의하는 능력까지 갖춰야 한다는** 점이었습니다. 이번 프로젝트에서 가장 어려웠던 부분은 **겉으로 드러난 불편함 뒤에 숨겨진 진짜 문제를 찾고, 그 문제가 실제로 해결할 만한 가치가 있는지 판단하는 일**이었어요.

문제 발견 →
문제 정의 → 문제 해결



기획적 사고
= Why에서 시작하는 문제해결 능력
= Problem Solver



Why = The Purpose

What is your cause? What do you believe?

Apple: We believe in challenging the status quo and doing this differently

How = The Process

Specific actions taken to realize the Why.

Apple: Our products are beautifully designed and easy to use

What = The Result

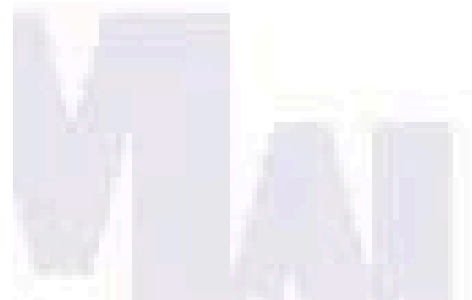
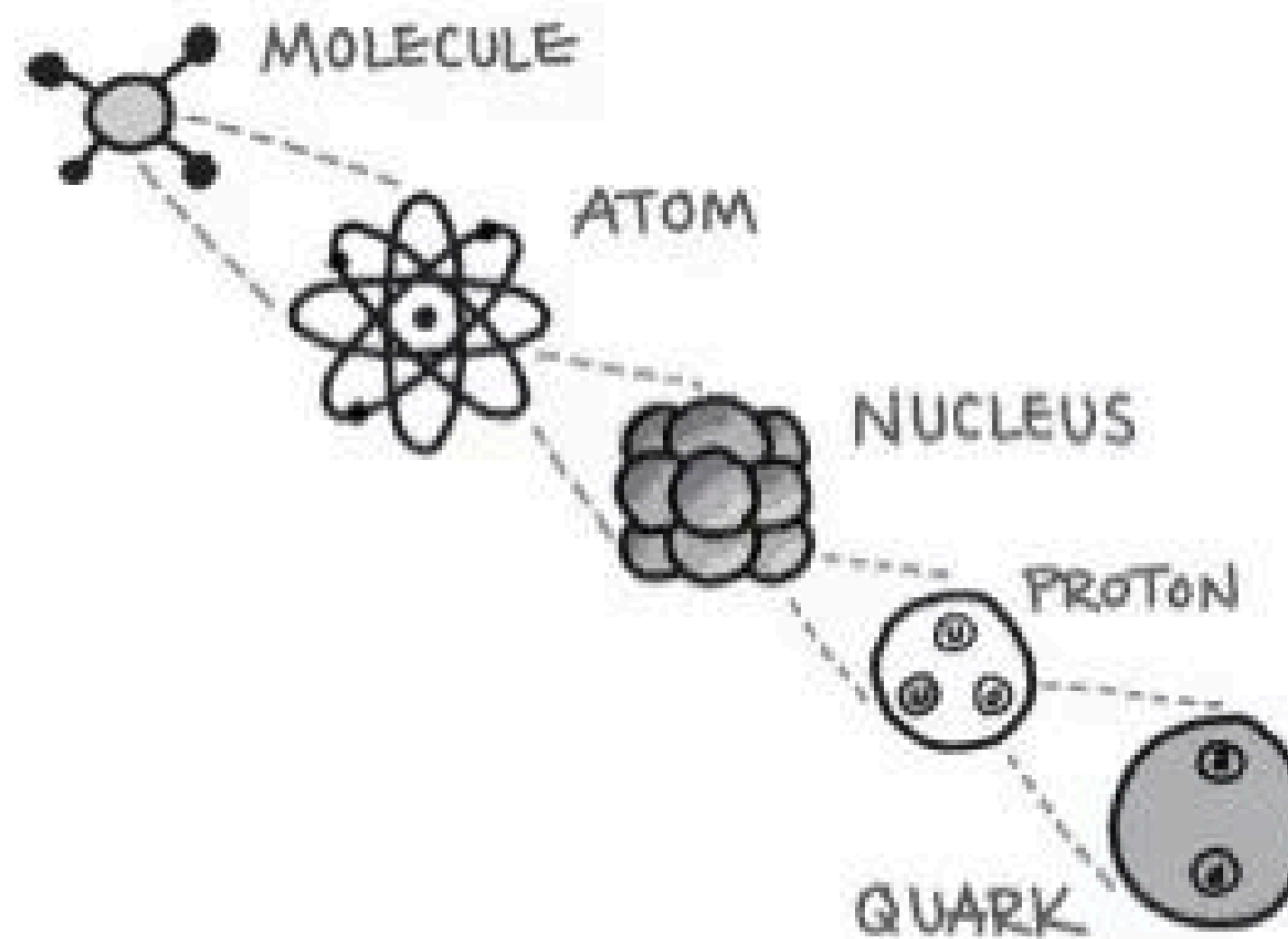
What do you do? The result of Why. Proof.

Apple: We make computers



제 1원칙 사고법 (First Principle Thinking)

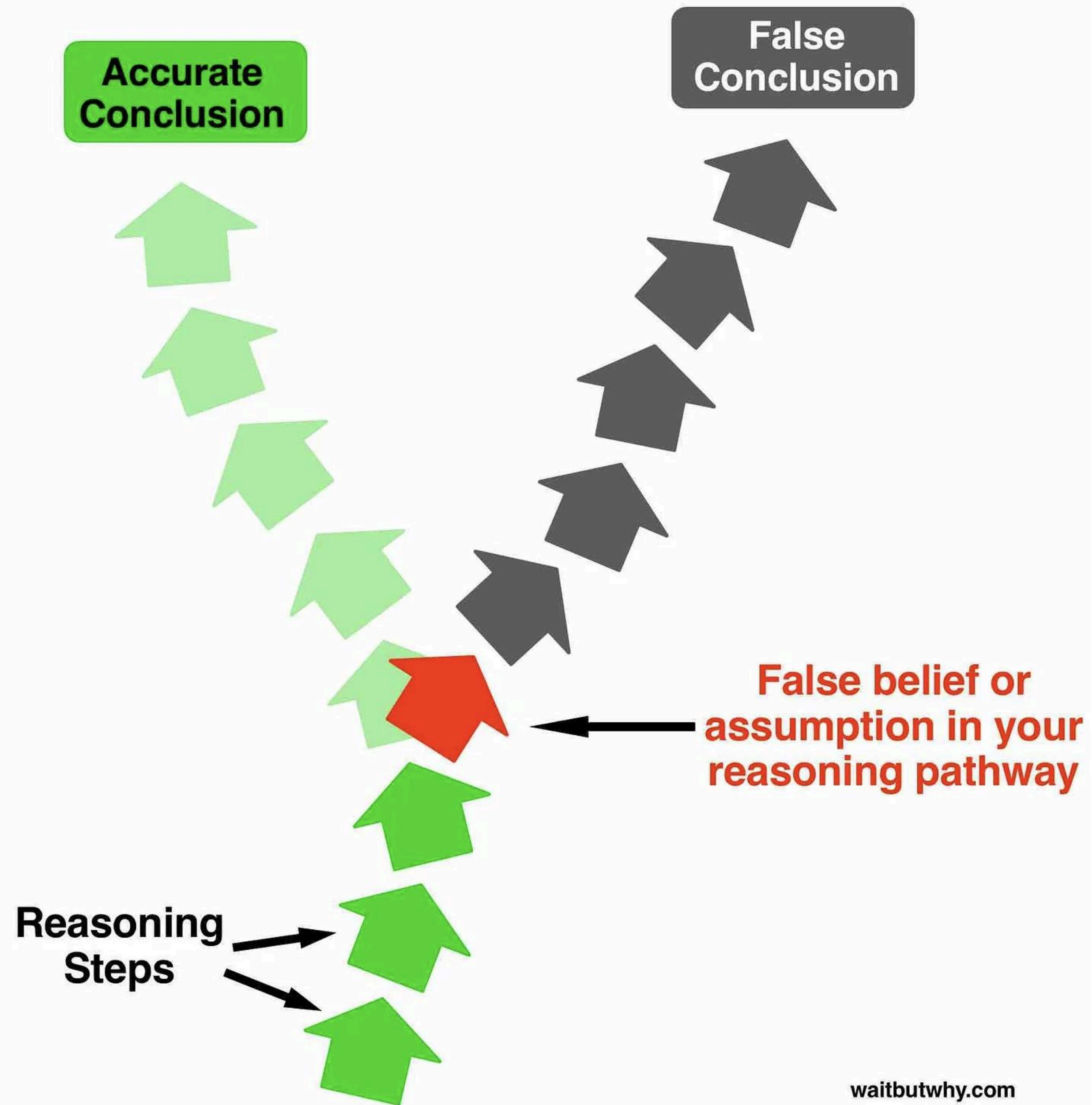
**제 1원칙 = 더이상 나눌 수 없는 가장
기본적인 사실이나 진리 (물리학)**



“Boiling problems down to their most fundamental truths.”

**문제를 가장 근본적인 요소로 분해하고,
그로부터 새로운 해결책을 구축해나가는 접근법**

**기존의 관행이나 유추에 의존하지 않고,
문제의 본질을 파고드는 것이 핵심**



SpaceX

- 기존 가정: “로켓은 원래 비싸다”



SpaceX

- 기존 가정: “로켓은 원래 비싸다”
- 제 1원칙 사고:
 - 로봇은 무엇으로 만들어졌는가?
 - 알루미늄 합금, 티타늄, 구리, 탄소 섬유 등
 - 이 원자재의 시장 가격은 어떻게 되는가?
 - 로켓 원자재 비용은 완성된 로켓 가격의 2%에 불과
 - 나머지 98%는 제조 과정의 비효율과 중간 마진
- 결론: 직접 원자재를 사서 만들면 비용을 획기적으로 줄일 수 있겠구나! → SpaceX

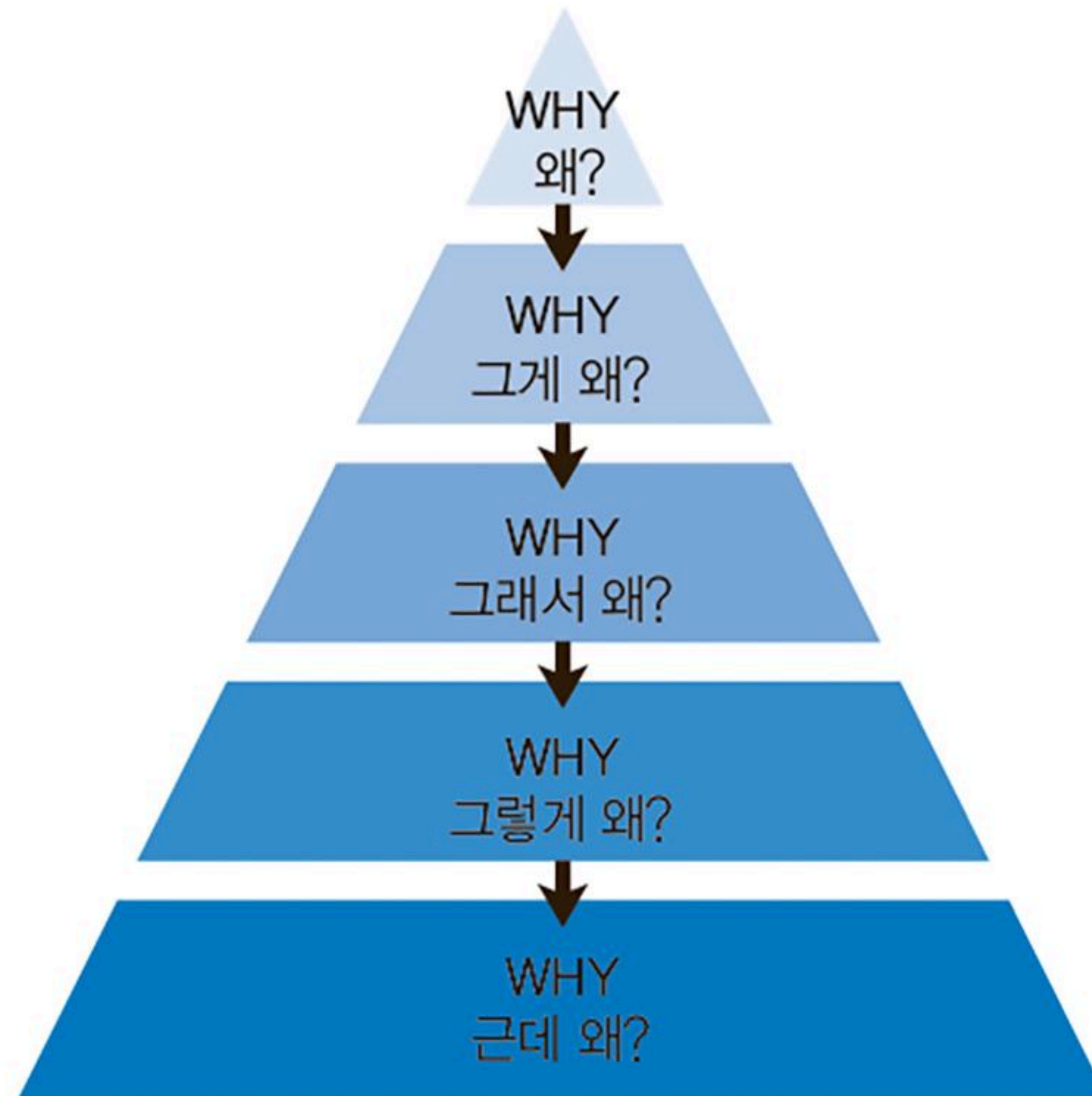
현상이 아닌
문제의 본질

5 Whys Method

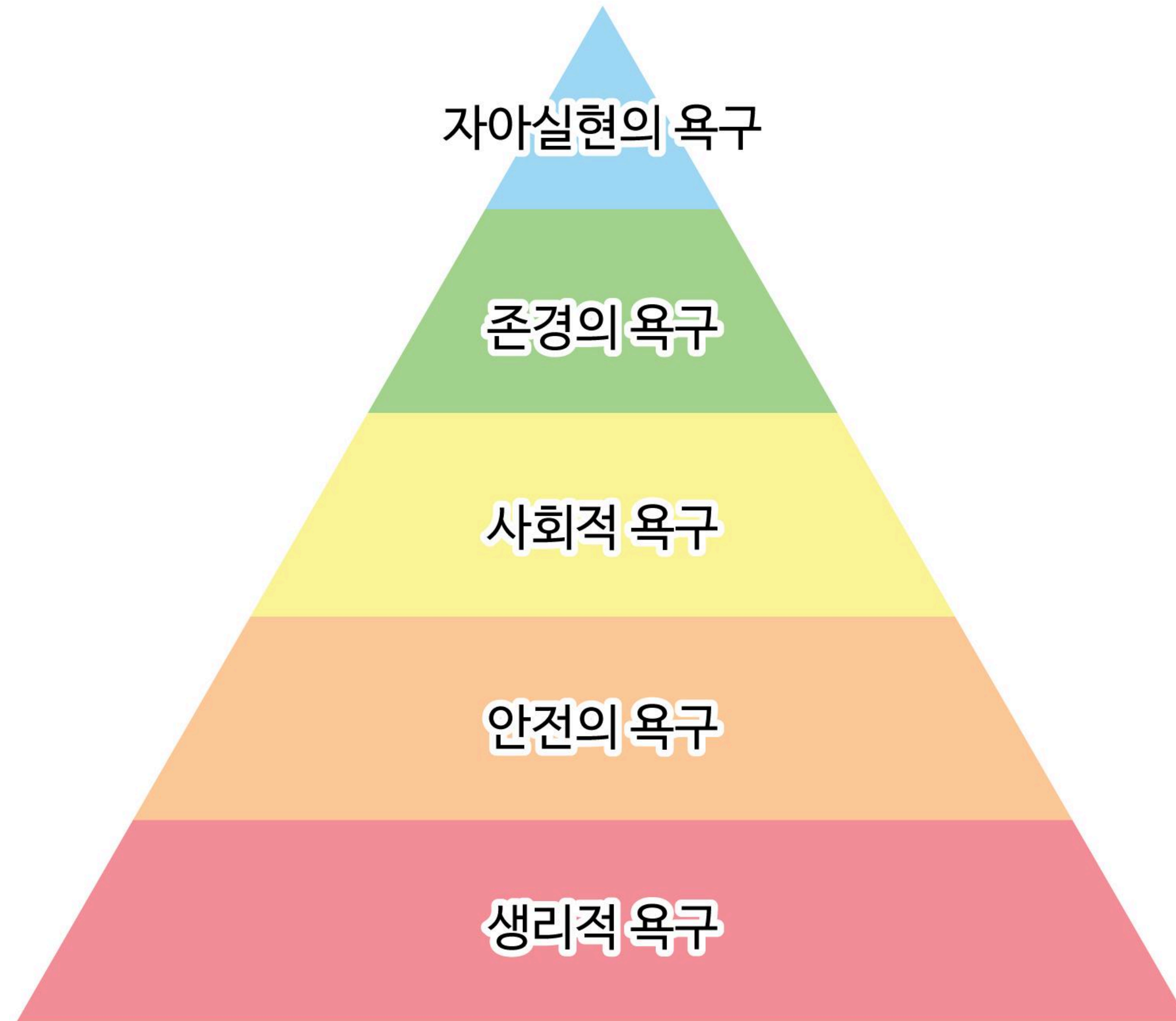
문제의 본질을 파헤칠 때까지
"왜"라고 5번 물어보기!

겉으로 드러난 문제와
진짜 문제는 다르다!

5WHY 개념도



때로는 문제의 본질까지 가보기



[매슬로우의 욕구 위계론]

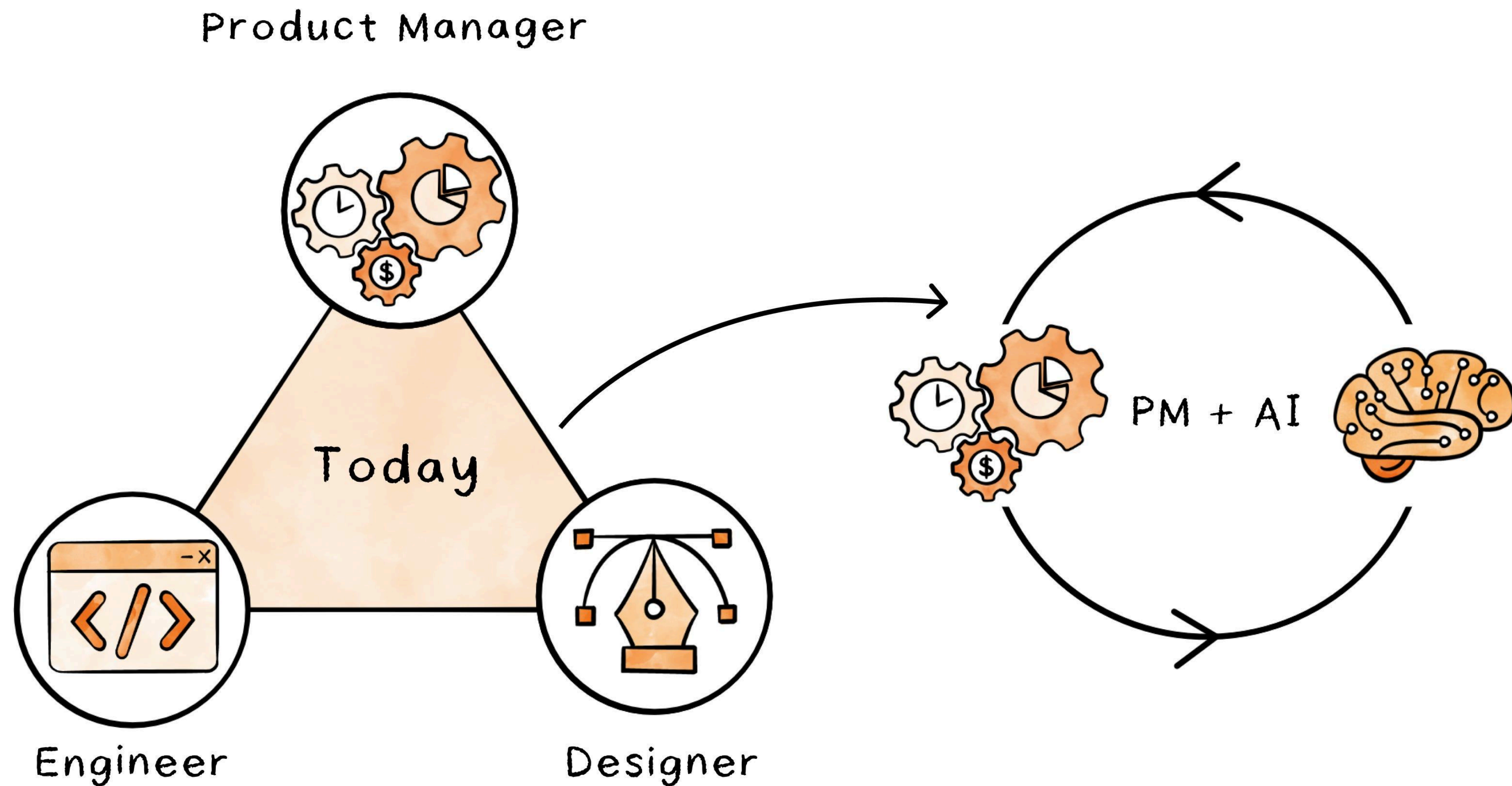
(AI) PM적 사고

PM = 지휘자





AI PM = Product Architecture

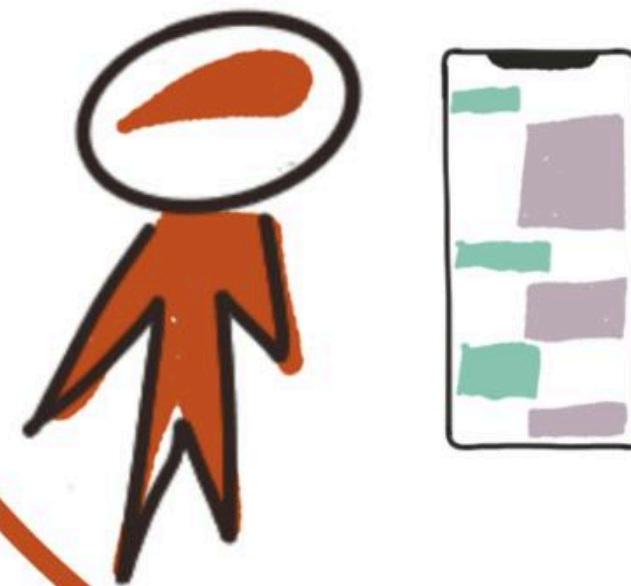


AI Curious PM ?

AI ENHANCED PM



AI EXPERIENCES PM



AI BUILDER PM



INSPIRED BY
DR. MARILY NIKA

AI PM skill set

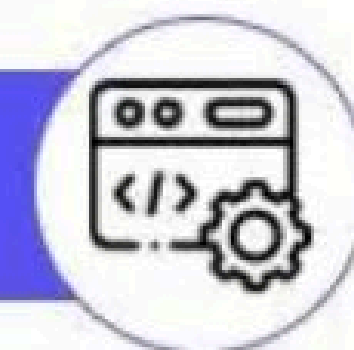
- 1 User empathy
 - 2 Business acumen
 - 3 Product/UX design
 - 4 Technical depth
 - 5 Communications
 - 6 Leadership
 - 7 Prioritization
 - 8 Data analysis
- ... and more

Traditional s/w PM skills



Additional skills for AI PMs

- 1 AI/ML technical foundation
 - 2 ML problem definition
 - 3 Data literacy
 - 4 Better management of uncertainty and ambiguity
 - 5 Risk management
 - 6 Explainability
 - 7 **Remain user-centric**
- ... and more



우리 프로덕트는 어떤 문제를 해결하려고 하는가?

풀 가치가 있는 진짜 문제를 찾는게 프로덕트 성공의 열쇠!

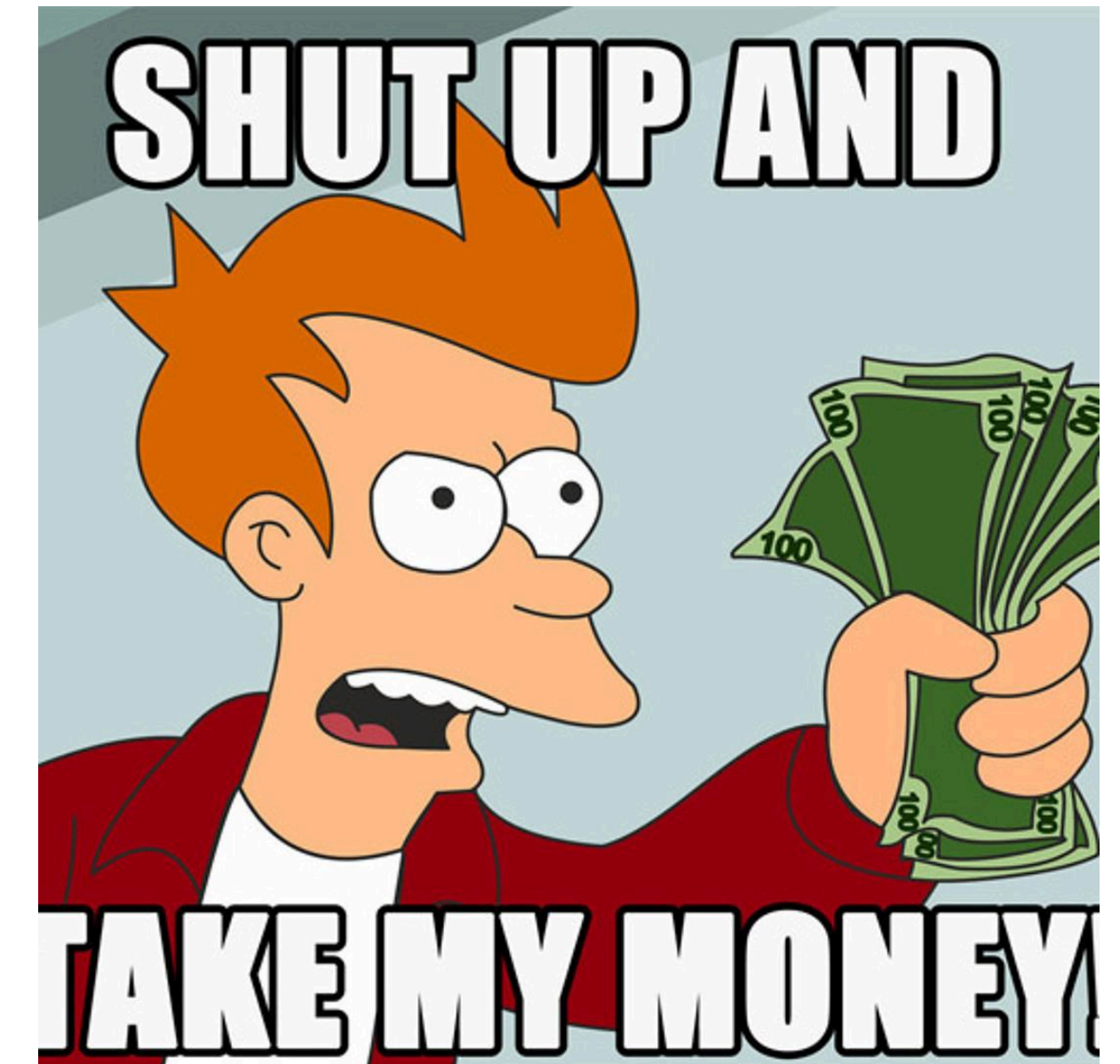
어떤 고객들이 어떤 어려움을 겪고 있지?

이 불편함이 얼마나 커서 돈을 얼마 낼 의향이 있지?

이 문제를 해결하면 우리 비즈니스에 어떤 도움이 되지?

고객이 돈을 지불하는 '4가지 효용'

1. **장소 효용:** 접근하기 어려운 걸 접근 가능하게 만들기
2. **형태 효용:** 기존의 것들을 재배열해 더 가치있게
3. **시간 효용:** 귀찮은 것, 느린 것을 빠르게 만들기
4. **소유 효용:** 중간 단계를 제거하기



모든 아이디어(문제)는 가설일 뿐이다.

→ 만약 ~ 한다면, ~하게 될 것이다

예) 만약 우리가 채팅봇을 도입한다면, 구매 전환이 10% 올라갈 것이다.



**Why & What
(PM)**

X

**How
(Dev)**

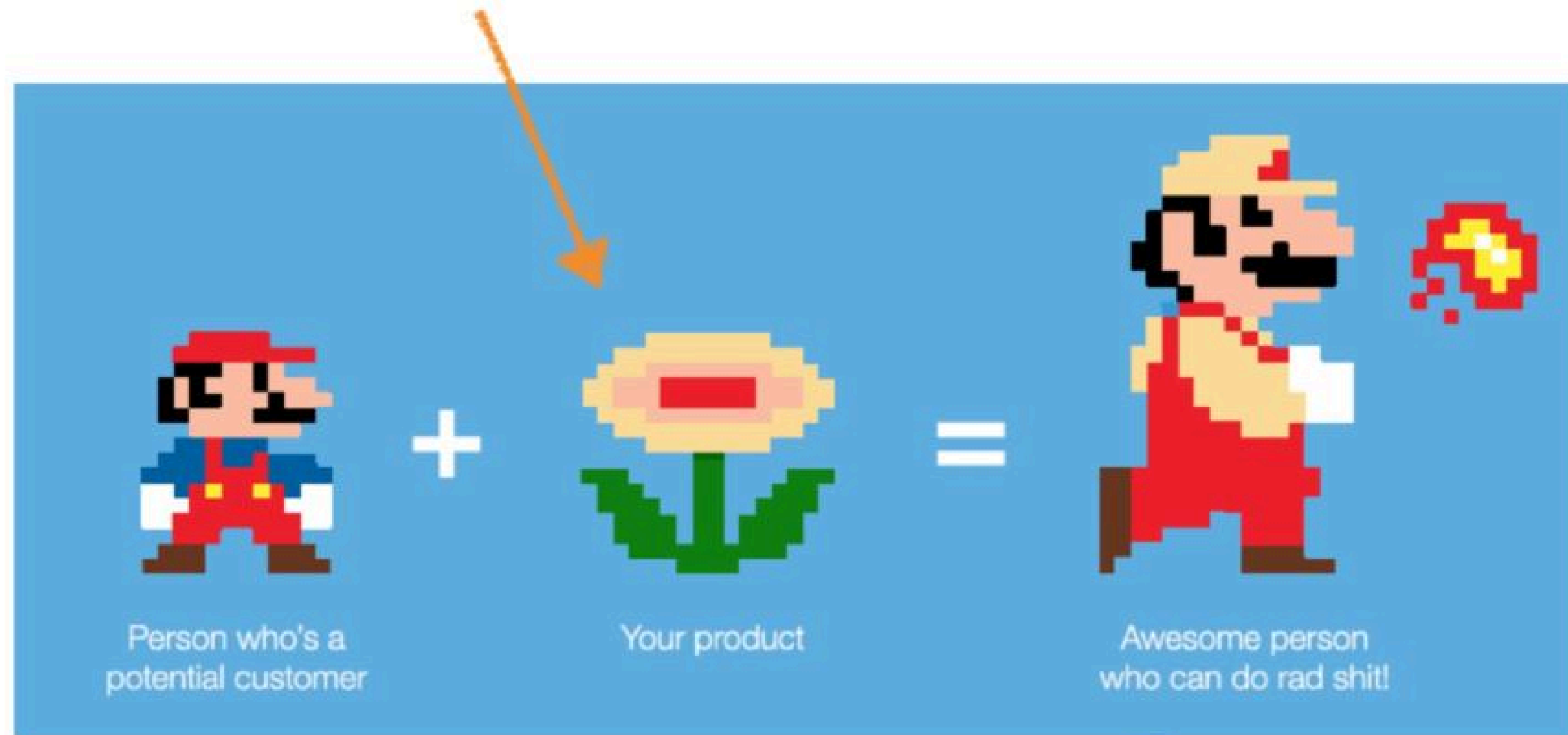
=

**최고의 제품
(Great Product)**

고객이 진짜 피부로 느끼는
“고통”에 집중하면,
“핵심 가치”가 보인다.

[고통받는 고객]이 [우리 제품]을 만나 [문제]를 해결했다!!

This isn't what your business makes



@UserOnboard

This is

AI 프로덕트란?

일반적인 프로덕트와 어떻게 다른가?

AI Product는
어떻게 다른가?

AI 프로덕트?
그거 챗지피티 아님?

LIMITED OFFER

NEW Kling 3.0 – Exclusive on Higgsfield

Kling 3.0 Exclusive Access For Everyone & 365 unlimited Nano Banana Pro. 52% OFF



Explore

Image

Video

Edit

Character

Contests

New

Vibe Motion

Cinema Studio 2.0

2.0

AI Influencer

Apps

Assist

Community

Pricing

Login

Sign up



CINEMA STUDIO 2.0

Step inside 3D scenes, control every shot, craft cinematic sequences



KLING 3.0 PUBLIC RELEASE

Kling 3.0 is now public with reduced wait times and faster generation



HIGGSFIELD CONTESTS

Submit your AI-generated action video and compete for \$500,000

Generate an AI action short and win \$500,000

Build a world, pick your hero, raise the stakes, and create a scene that makes people wish your film was real

Enter now



[Pricing](#)[Chrome extension](#)[Get started](#)

Transform Messy Sources into Actionable Insight

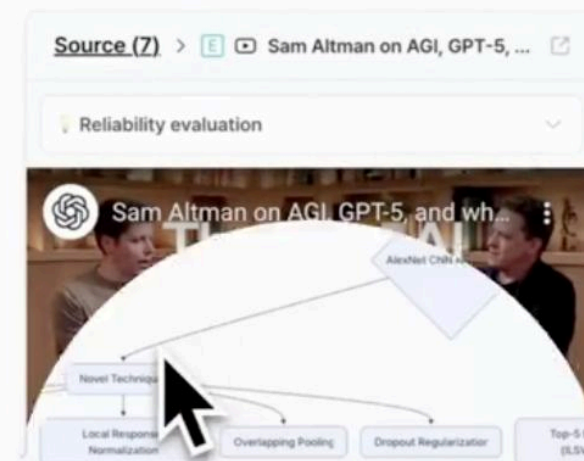
Turn videos, PDFs, and websites into AI-powered insight reports and source-grounded chat—built for leaders, researchers, learners.

[Start for free](#)

📈 1.2M monthly visits (Similarweb)

👤 2.6K Product Hunt followers

📄 8.5M reports created



Download for macOS 

The screenshot displays the Cursor IDE interface. On the left is a sidebar with a task list under the heading 'IN PROGRESS 1'. The tasks are: 'Analyze Tab vs Agent Usag...' (Fetching data), 'Build Landing Page' (now Done. Fonts preload in the he...), 'Plan Mission Control' (now +20 -3 · Drafted implementati...), 'PyTorch MNIST Experi...' (10m PyTorch MNIST Experiments), 'Set up Cursor Rules fo...' (30m Set up Cursor Rules for Dashb...), and 'Bioinformatics Tools' (45m +135 -21 · Bioinformatics Tools). The central workspace shows a file named 'feature-prd.md' with a thought: 'let's build a mission control interface, similar to the expose-style window manager on macOS'. Below this, it says 'Thought 4s', 'Read AppManager.tsx', and 'Searched expose patterns'. A draft implementation step is shown: 'Drafted implementation steps in feature-prd.md. A few quick questions before I start building:'. The right sidebar has a tab for 'feature-prd.md' and a 'Build' button. It contains a section titled 'Mission Control Interface' with the text: 'A grid view of all open windows as scaled previews, allowing quick selection to bring any window to front.' Below this is a 'Trigger' section: 'Menu item in MenuBar.tsx (View > Mission Control), hotkey F3, or double-tap desktop.' and a 'View Behavior' section: 'Overlay existing windows into a grid of live previews with spring-based layout animations and shared element transitions.' At the bottom, there is a '3 Tasks' section with a list of tasks: 'Add multiplayer mode to useAppStore.ts', 'Create a new MissionControlView.tsx component', 'Update AppManager.tsx to apply expose modes.', and 'Add a task. ⌘K to generate...'.

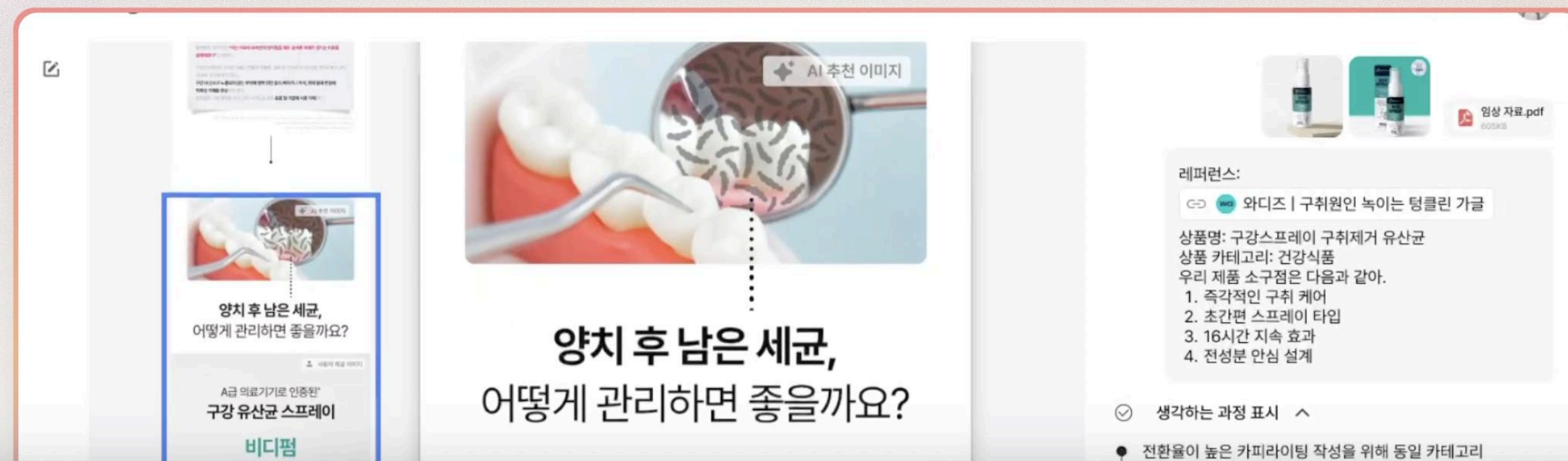
20K+ 개 이상의 이커머스 상세페이지가 후커블로 생성됩니다

압도적 퀄리티 상세페이지, AI로 5분만에 완성.

제품 정보 30초 입력하고, 지금 바로 다운로드 하세요.

지금 시작하기

G ★★★★★ | 실사용자 사용 후기 평점 4.9점



일반적인 프로덕트 vs. AI 프로덕트

Traditional s/w products	vs	AI/ML Products
Meet a functional spec	Success Criteria	Performance accuracy
S/W code only	Quality depends on	S/W code + data + parameters
One or two S/W stacks	Tools	Many algorithms/libraries/frameworks
SWE	Developers	DE+DS+MLE+SWE
Forward + incremental	Dev progress	Exploratory & non-incremental
Easier to estimate	Dev timeline	Hard to estimate
Relatively easy	Interpretability	Difficult

Product vs. AI Product

일반 프로덕트	AI 프로덕트
결정론적(deterministic) • $1+1 = 2$ • if A then B • 결과 예측 가능, 오류는 버그	확률론적(probabilistic) • 매번 답변이 살짝 달라짐 • "가장 정답일 확률이 높은 토큰"을 생성함
고정된 로직 • 제품 기능이 코드에 고정 • 기능 바꾸려면 코드 업데이트	데이터 기반의 진화 • 코드보다 데이터와 모델의 가중치가 제품의 성능을 결정, RAG나 프롬프트로 제품 성능 비약적 개선 가능; 피드백 루프
명령 기반 UI • 사용자가 정해진 인터페이스에 적응 • 버튼 누르면 → 특정 행동	의도 기반 UX (Intent-based, Semantic) • 제품이 사용자의 언어와 의도 이해 • AI가 알아서 필요한 기능을 호출함

Product vs. AI Product

일반 프로덕트	AI 프로덕트
QA (품질보증) • Pass/Fail가 명확한 유닛 테스트 • 예측된 입력값 → 예측된 출력값	Evals (평가) • 결과가 매번 달라 QA 불가 • 수백개의 질문 세트 던져 유사도나 안전성을 기반으로 하는 Eval 시스템 필수
선형적 비용 • 서버 호스팅 비용은 사용자 수에 비례해 어느 정도 예측 가능 • 한 번 개발하면 보수 비용 안정적	가변적 토큰 비용 • 사용자가 말할때마다 모델 API 및 토큰(Token) 비용 발생, 모델이 좋을수록 비용이 기하급수적으로 비싸짐 • BM과 추론비용(Inference Cost) 계산
“사용자 문제를 해결하기 위해 어떤 기능을 만들까?”	“어떤 데이터를 확보해서, 모델이 사용자의 의도를 얼마나 높은 확률로 해결하게 만들까?”

if-then (만약 A면 B를 해라)
로직으로 100% 구현 가능하다면?

→ **AI는 비용이 비싸고, 느리고,
통제하기 어려움; 굳이 X**

OpenAI도 적자인데 우리가 뭘....

AI • MARKETS

OpenAI won't make money by 2030 and still needs to come up with another \$207 billion to power its growth plans, HSBC estimates



By Nick Lichtenberg
Business Editor

November 26, 2025, 1:24 PM ET

[Add us on](#)



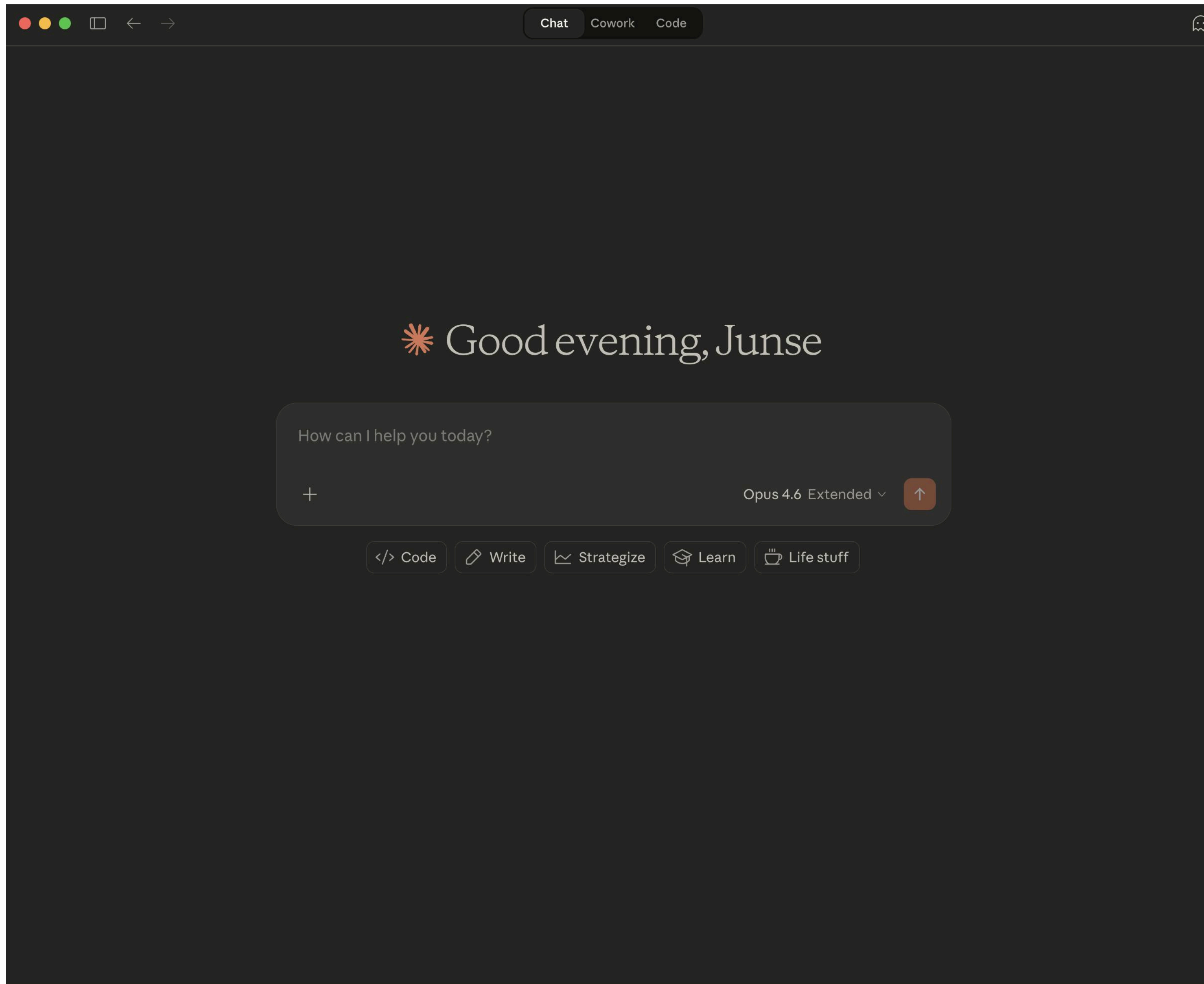
OpenAI CEO Sam Altman

NATHAN HOWARD—BLOOMBERG/GETTY IMAGES

AI 제품은 3종류만 있다 (아직)

1. 챗봇 (Chatbot)
2. 자동 완성 (Github Copilot)
3. 에이전트 (AI Agents)





```
<div className="grid-cols-2 gap-4 mb-4">
  <div className="bg-white rounded-xl p-4 shadow-sm">
    <h3 className="font-semibold text-gray-700 text-sm mb-3">📊 과목별 매출 비중</h3>
    <div className="flex items-center">
      <ResponsiveContainer width="55%" height={180}>
        <PieChart>
          <Pie data={subjectTotal} cx="50%" cy="50%" innerRadius={40} outerRad
            {subjectTotal.map((entry, index) => (
              <Cell key={`cell-${index}`} fill={entry.color} />
            ))}
          </Pie>
          <Tooltip formatter={(value) => `${(value / 100000000).toFixed(1)}억`>
        </PieChart>
      </ResponsiveContainer>
      <div className="w-[45%] pl-2">
        {subjectTotal.map((item, index) => (
          <div key={index} className="flex items-center justify-between py-1">
            <div className="flex items-center">
              <div className="w-2.5 h-2.5 rounded-full mr-1.5" style={{ backgr
                <span className="text-xs text-gray-600">{item.name}</span>
            </div>
            <span className="text-xs font-semibold text-gray-800">
              {(item.value / totalSales) * 100).toFixed(1)}%
            </span>
          </div>
        ))}
      </div>
    </div>
  </div>
</div>
```

```
<div className="bg-white rounded-xl p-4 shadow-sm">
  <h3 className="font-semibold -gray-700 text-sm mb-3">17 요일별 매출 (백만원)
  <ResponsiveContainer width="100%" height={180}>
```

Reply...



Opus 4.5 Extended ▾

mylight-agent — drive.py

Open Agent Manager

Log in with Google →

Explorer

mylight-agent

- .claude
- agents
 - 01_trend_researcher.md
 - 02_ideator.md
 - 04_instructor_finder.md
 - 06_planner.md
 - 07_email_drafter.md
- api
 - __pycache__
 - __init__.py
 - drive.py
 - gmail.py
 - google_auth.py
 - sheets.py
- config
 - company_profile.md
 - competitors.md
 - lecture_format.md
- outputs
- templates
- .gitignore
- CLAUDE.md
- credentials.json
- README.md
- token.json

drive.py

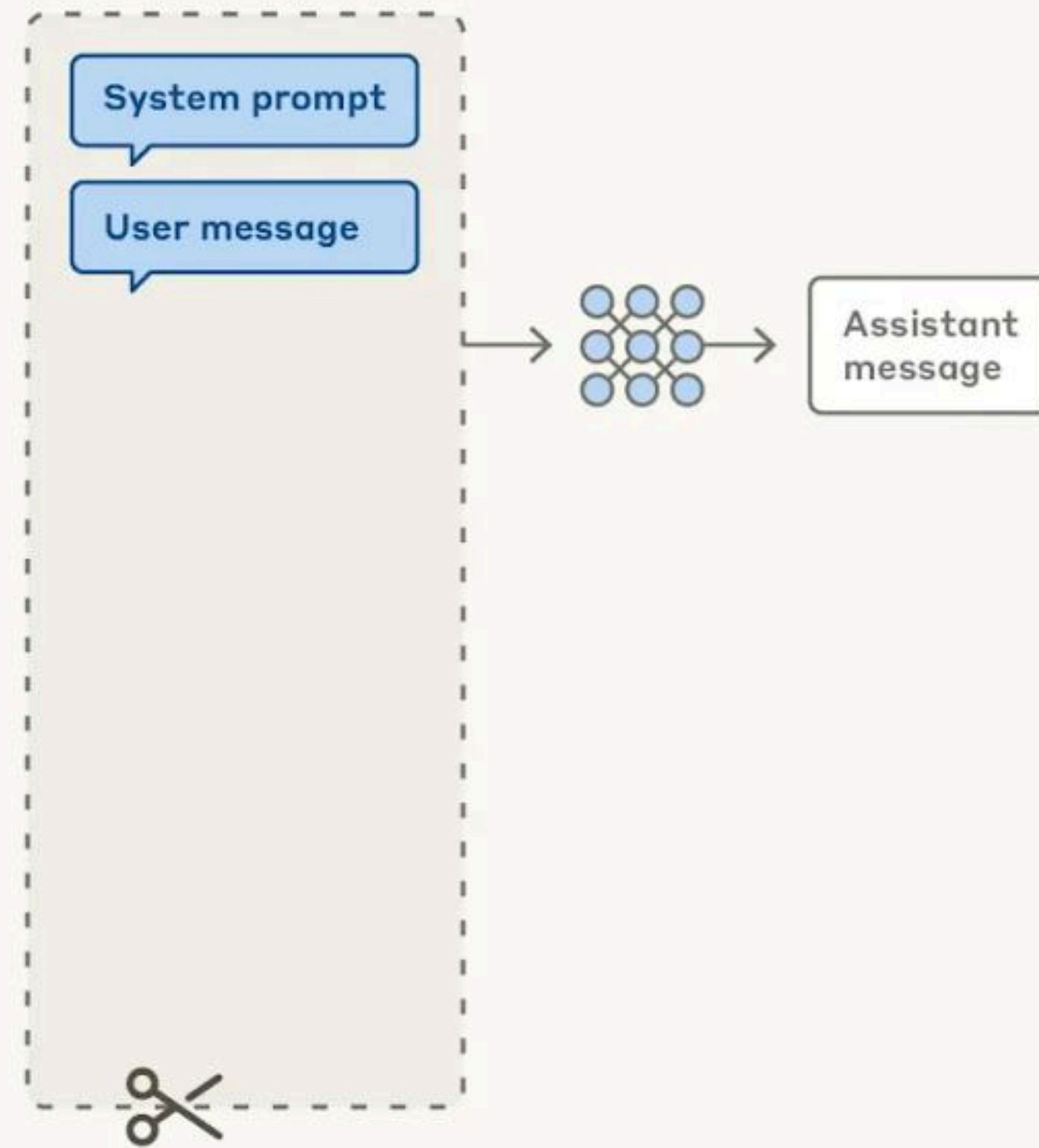
api > drive.py > ...

```
7 사전 설치:
8 | pip install google-api-python-client google-auth
9 |
10
11 import sys
12 from pathlib import Path
13 from googleapiclient.discovery import build
14 from googleapiclient.http import MediaFileUpload
15
16 PROJECT_ROOT = Path(__file__).parent.parent
17 OUTPUTS_DIR = PROJECT_ROOT / "outputs"
18
19 # 드라이브 폴더명
20 DRIVE_FOLDER_NAME = "마이라이트 기획안"
21
22 | %I for Command, %L for Agent
23 def get_service():
24     """Google Drive API 서비스 반환"""
25     from google_auth import get_oauth_credentials
26     creds = get_oauth_credentials()
27     return build("drive", "v3", credentials=creds)
28
29
30 def get_or_create_folder(service, folder_name):
31     """드라이브 폴더 생성 또는 가져오기"""
32     # 기존 폴더 검색
33     query = f"name='{folder_name}' and mimeType='application/vnd.google-apps.folder' and trashed=false"
34     results = service.files().list(q=query, fields="files(id, name)").execute()
35     files = results.get("files", [])
36
37     if files:
38         return files[0]["id"]
39
40     # 폴더 생성
41     file_metadata = {
42         "name": folder_name,
43         "mimeType": "application/vnd.google-apps.folder",
44     }
45     folder = service.files().create(body=file_metadata, fields="id").execute()
46     print(f"새 폴더 생성: {folder_name}")
47     return folder["id"]
48
49
```


Prompt engineering vs. context engineering

Prompt engineering for single turn queries

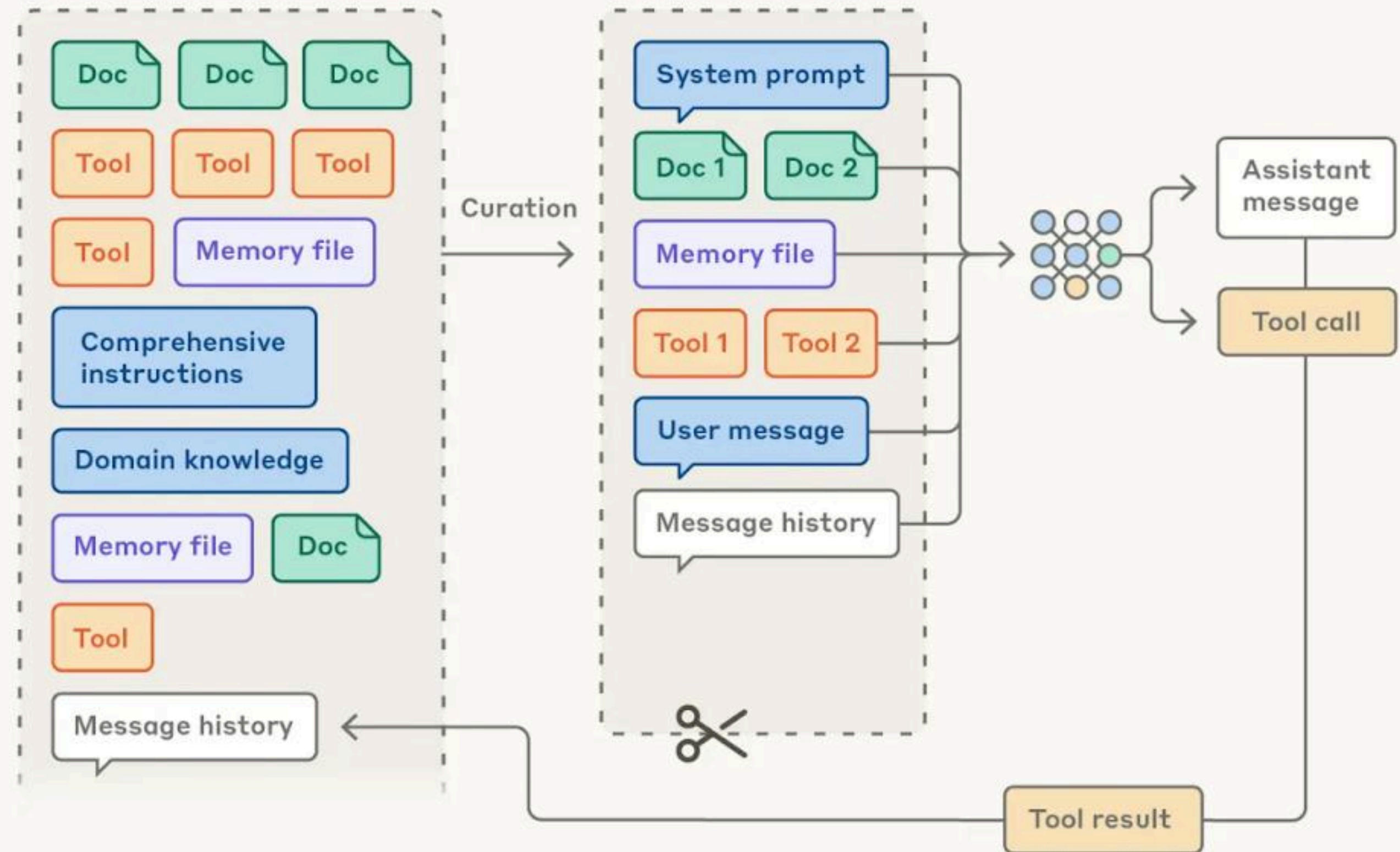
Context window



Context engineering for agents

Possible context to give model

Context window



에이전트 엔지니어링

**프롬프트 + 도구 + 권한 + 구조 + 스킬을
종합적으로 설계하고 반복 개선하는 것**

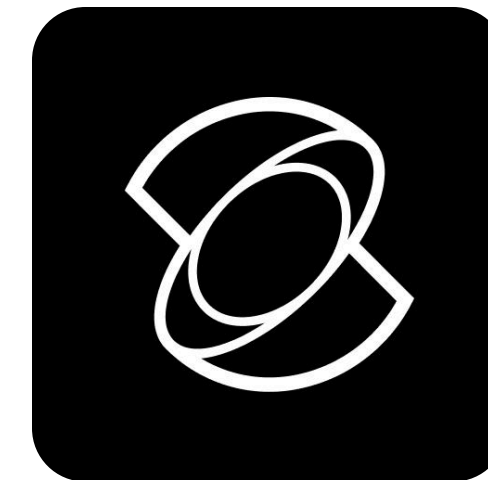
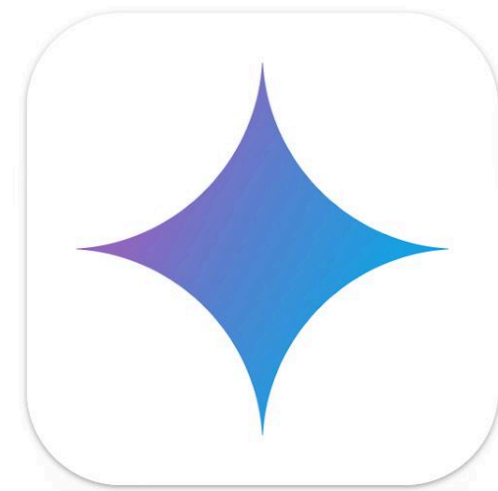
AI PM 핵심 역량

- 문제를 잘 쪼개는 능력: 이 작업을 에이전트 몇 개 필요?
- 좋은 지시를 쓰는 능력: 잘 쓴 프롬프트가 곧 답변 퀄리티
- 테스트하고 원인을 찾는 능력: 왜 이상하게 동작했지?

AI 기본 용어

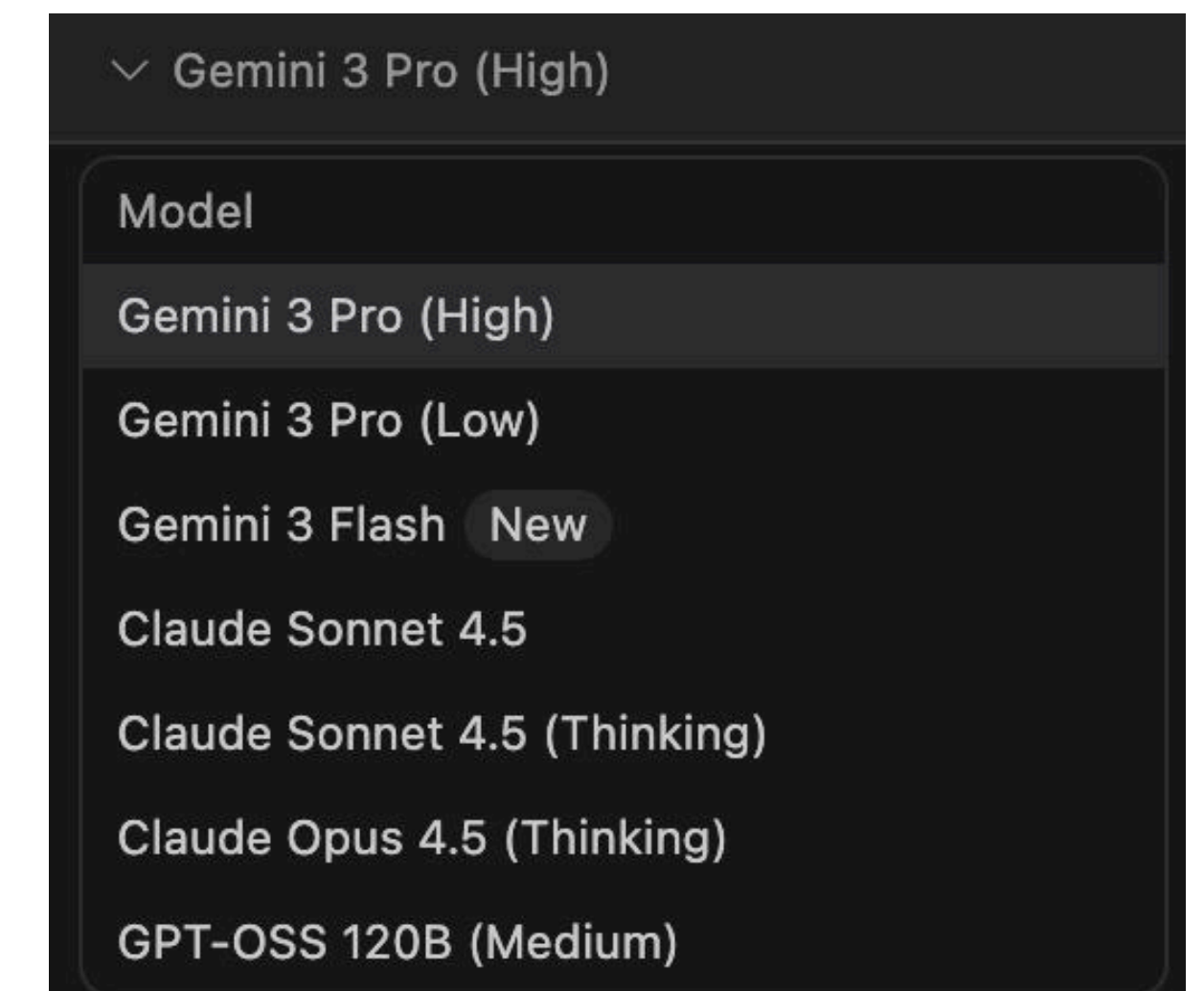
AI 기본 용어

- **Gen AI (Generative AI), 생성형 AI**
 - 텍스트, 이미지, 코드, 음악 등 새로운 콘텐츠를 '생성'할 수 있는 모든 AI 기술을 통칭하는 넓은 범위 용어
 - ChatGPT, Claude, Gemini → Gen AI에 포함
 - Midjourney, Sora → Gen AI **O**, but LLM **X**



AI 기본 용어

- **LLM(Large Language Models), 대규모 언어 모델**
 - 방대한 텍스트 데이터를 학습해 인간의 언어(자연어)를 이해하고 생성하는 **지능형 엔진** 혹은 두뇌 그 자체를 말함
 - ChatGPT 안에는 GPT-4o 라는 LLM 엔진 사용
 - Claude 안에는 Claude 3.5 Sonnet LLM 엔진 사용

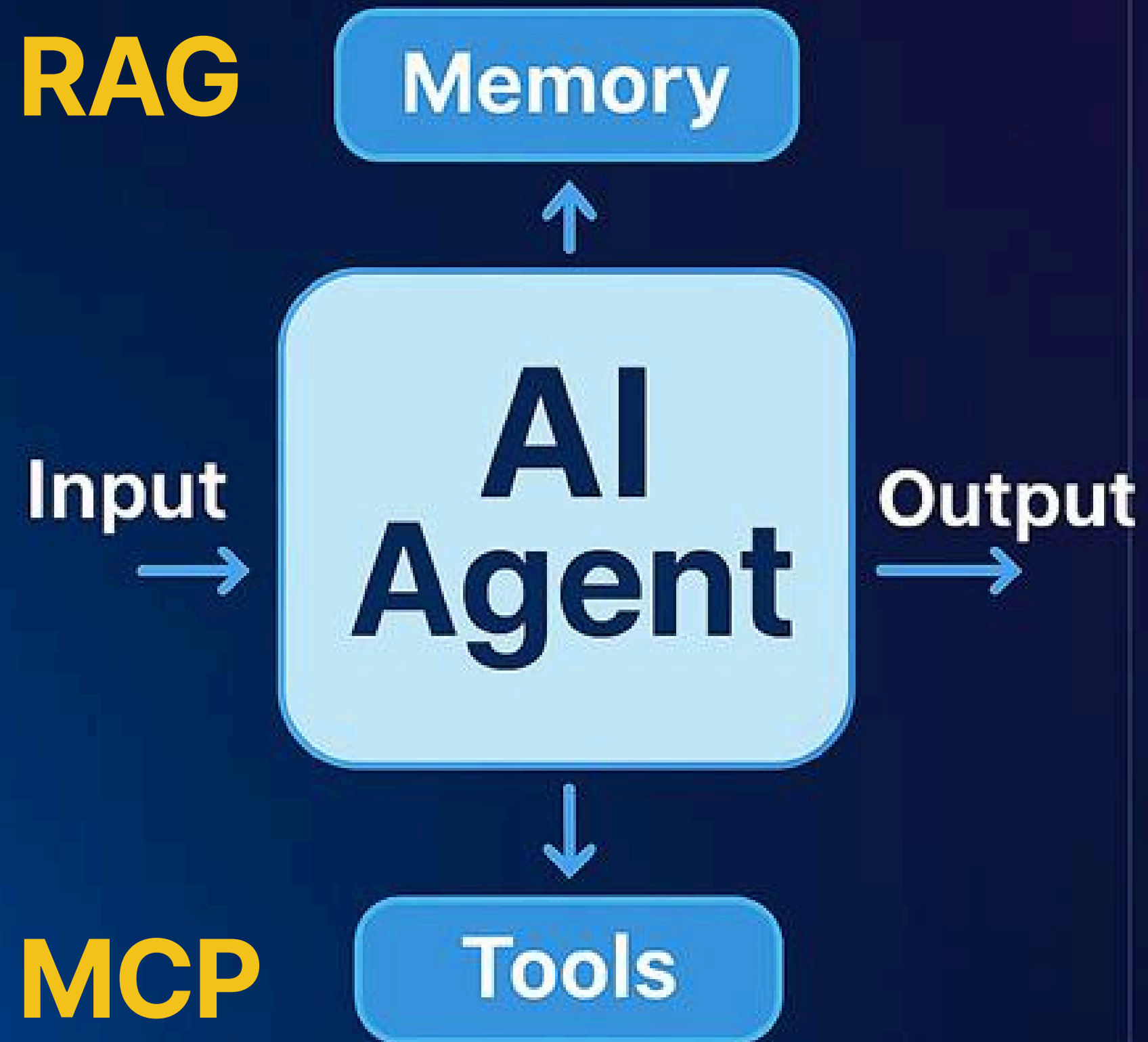


AI 기본 용어

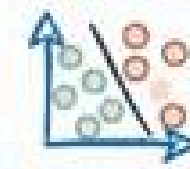
- **AI Agent**

- LLM = 고성능 엔진, 질문에 답하는 '두뇌'
- AI Agent = 엔진 장착하고 스스로 운전해서 목적지까지 가는 '**자율주행차**'
- 차이점
 - 챗봇에게 "제주도 맛집 알려줘"하면 리스트만 뽑아줌
 - 에이전트에게 "제주도 맛집 찾아서 내 캘린더 비는 시간에 예약까지 해줘"라고 하면 직접 맛집 검색하고 내 일정표 확인 후 예약 사이트에서 결제까지 시도

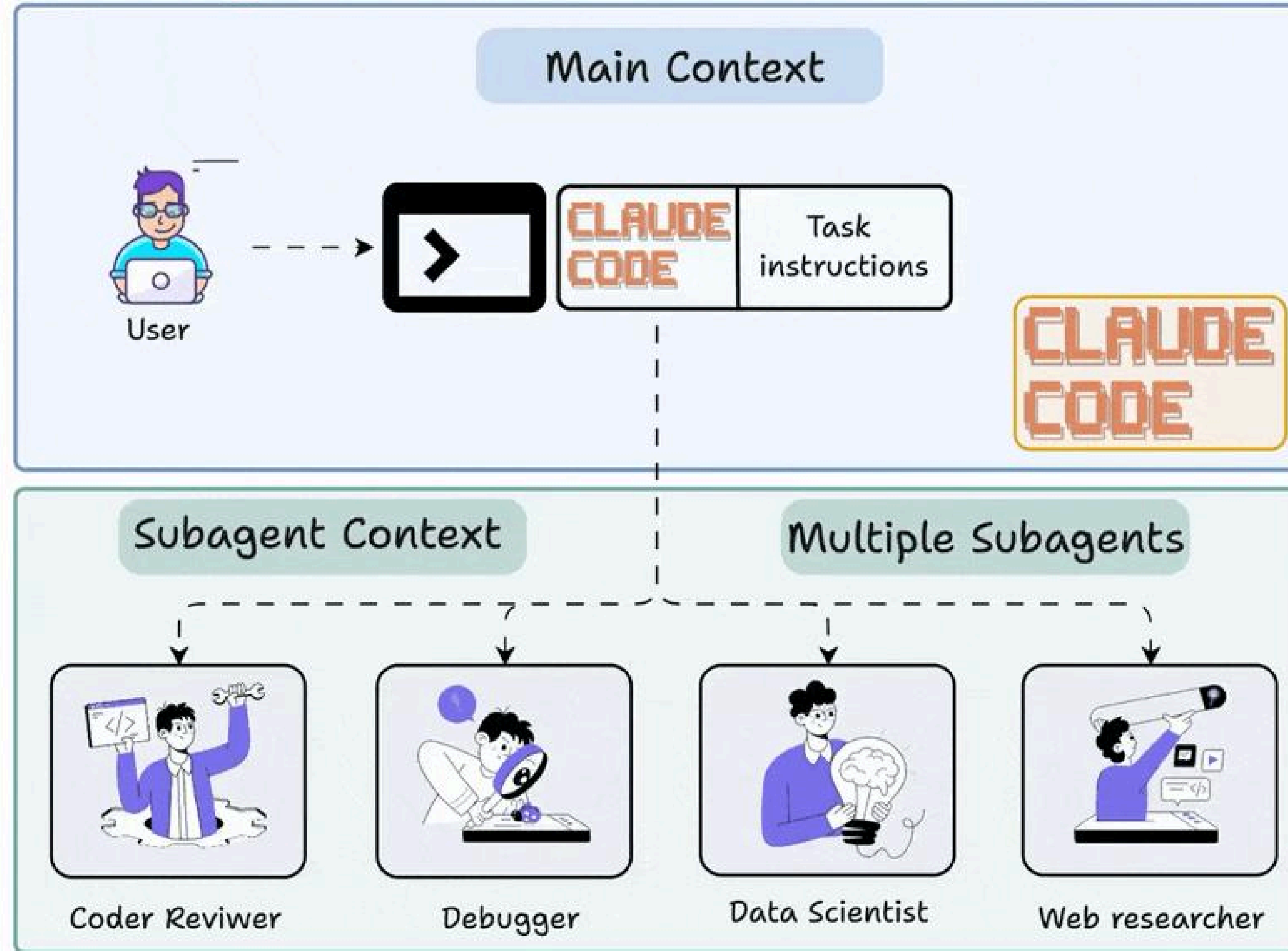
Gen AI Agent vs. LLM Flow



Subagents in Claude Code



DailyDoseOfDS.com



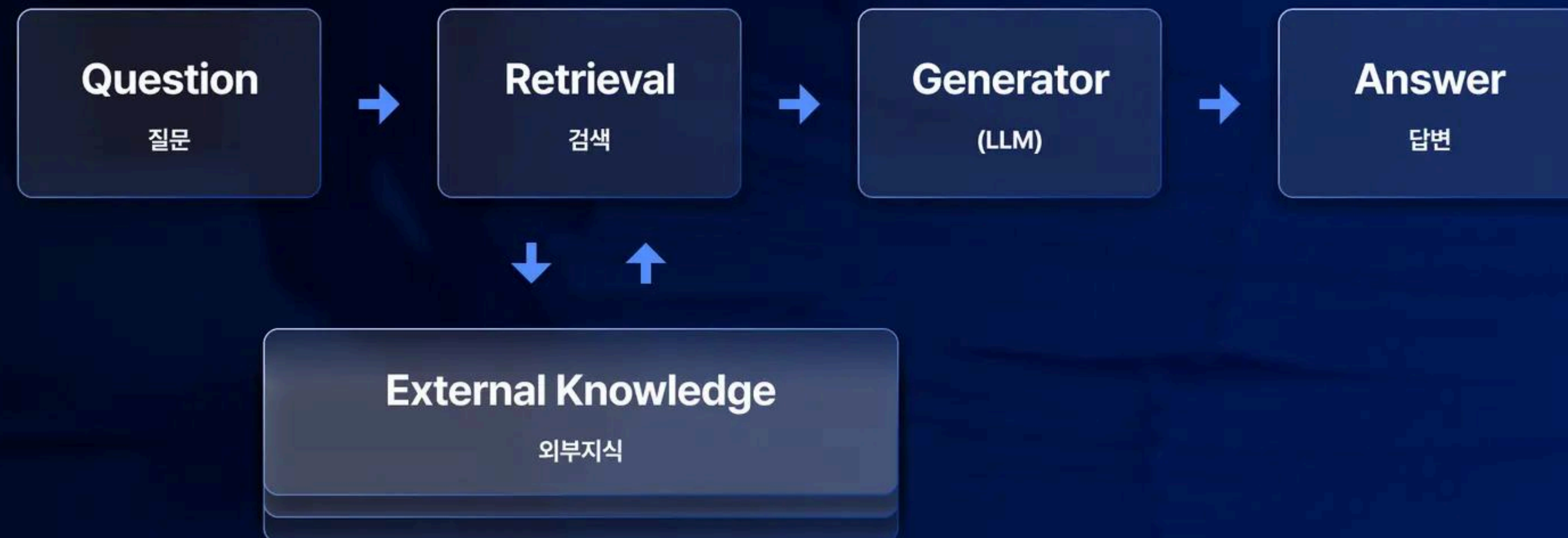
AI 기본 용어

- **RAG 검색증강생성**
 - 외부의 최신 정보를 실시간으로 가져와 엔진에게 전달해주는 기술; Retrieval-Augmented Generation
 - 질문이 들어오면 모델이 아는대로 답하기 전에, 먼저 **외부 데이터베이스(도서관)**에서 관련 내용을 “검색” 후, 그 내용을 바탕으로 답변을 “생성”함
 - 실시간으로 도로 정보를 알려주는 내비게이션 느낌
 - 할루시네이션 현상 획기적으로 줄어듦, 기업 전문직 사용

LLM



RAG

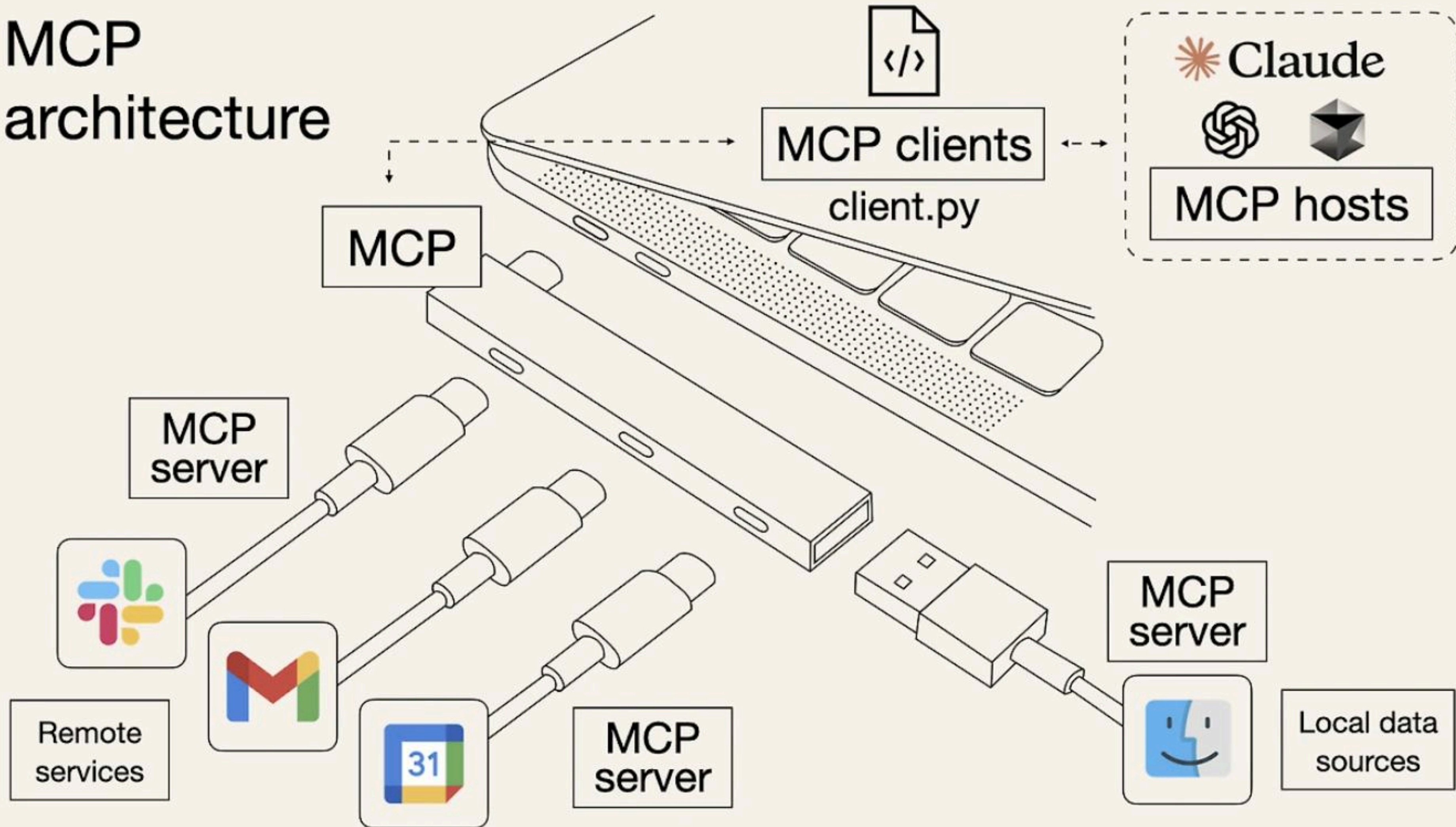




AI 기본 용어

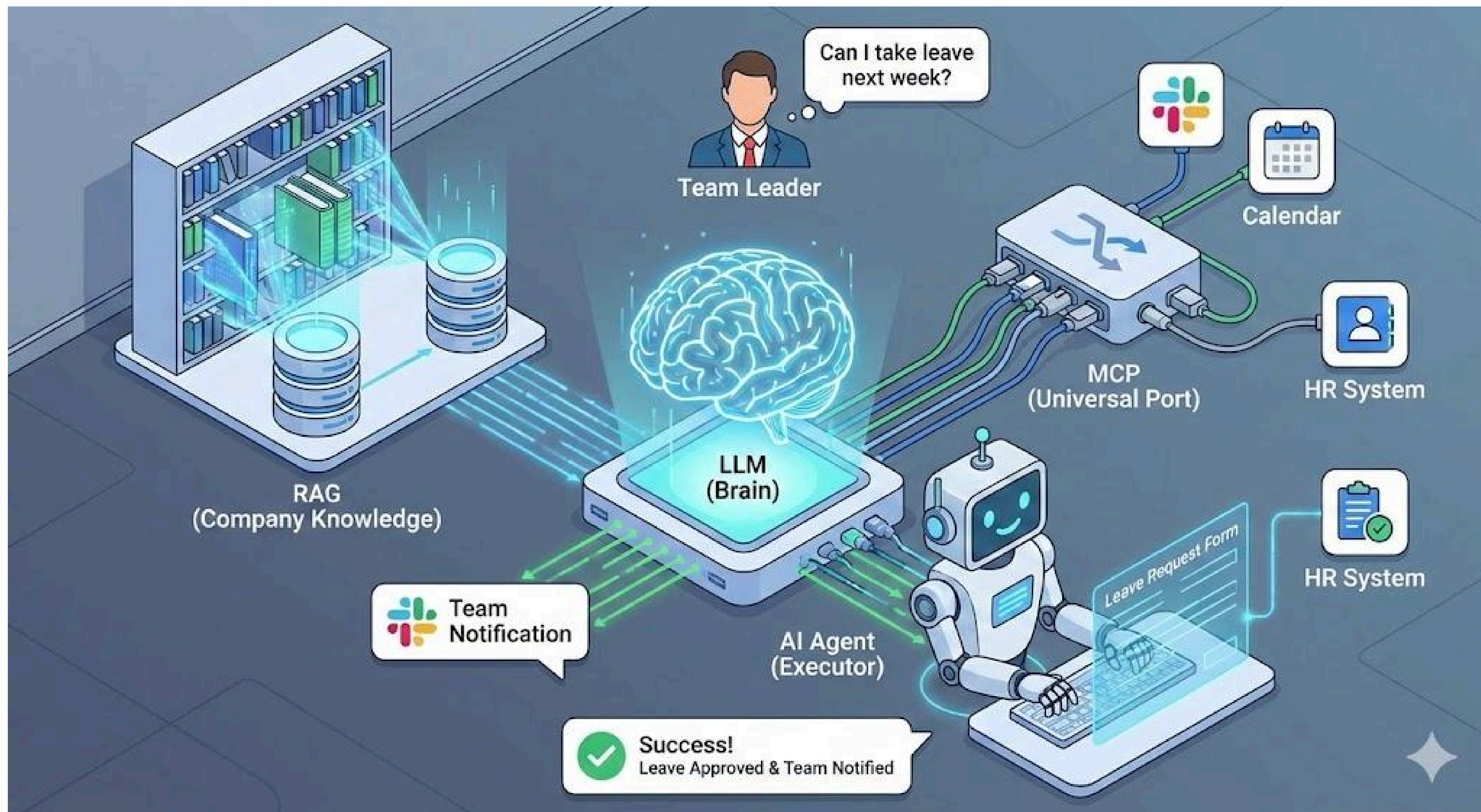
- **MCP (Model Context Protocol)**
 - 서로 다른 도구(소프트웨어)끼리 연결해주는 통일된 연결방식
 - 예전에는 AI에게 회사 내부 문서를 읽게 하려면 복잡한 코딩 (API 연동)이 필요했음.
 - 이제 데이터 제공자가 MCP 표준만 지키면, 어떤 AI 에이전트든 즉시 그 데이터를 이해하고 사용가능
 - 차에 스마트폰도 꽂고, 노트북도 충전하고, 블랙박스도 연결할 때 쓰는 **"범용 USB-C 포트"**처럼 연결의 규격을 통일

MCP architecture



AI 기본 용어

- **Skills.md**
 - 스킬 = 메뉴얼/레시피, "이럴 때는 이렇게 해라"는 마크다운 파일
 - 에이전트(서브에이전트) = 직원. 일 맡겨 독립적으로 처리하는 존재
- 같은 방식이 반복적으로 필요할 때 사용
 - 예시:
 - 우리 팀 코드 스타일 가이드
 - 문서 작성할 때 따라야 할 포맷
 - 코드 리뷰할 때 체크해야 할 항목
 - 엑셀/워드 파일 만드는 방법

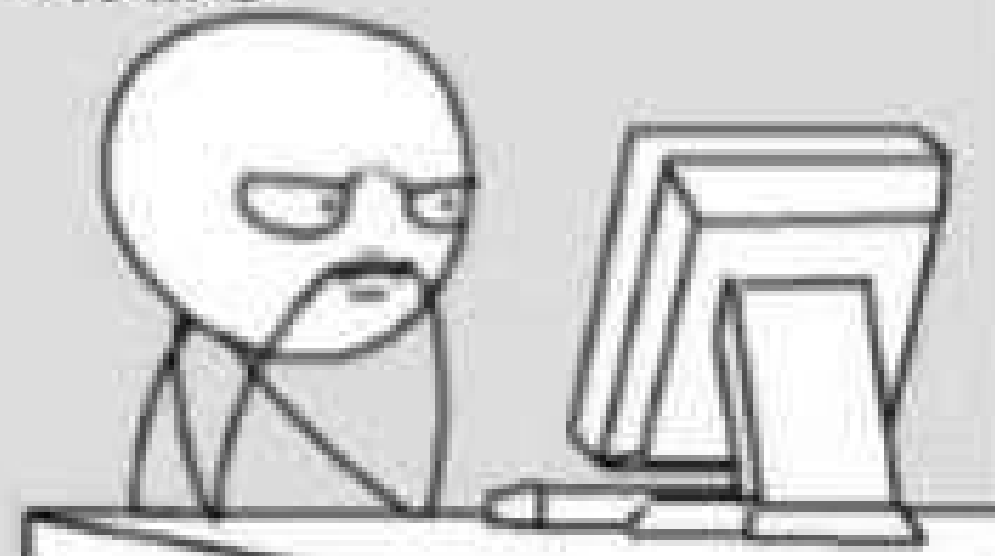


AI를 언제 활용할까?

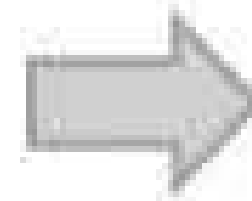
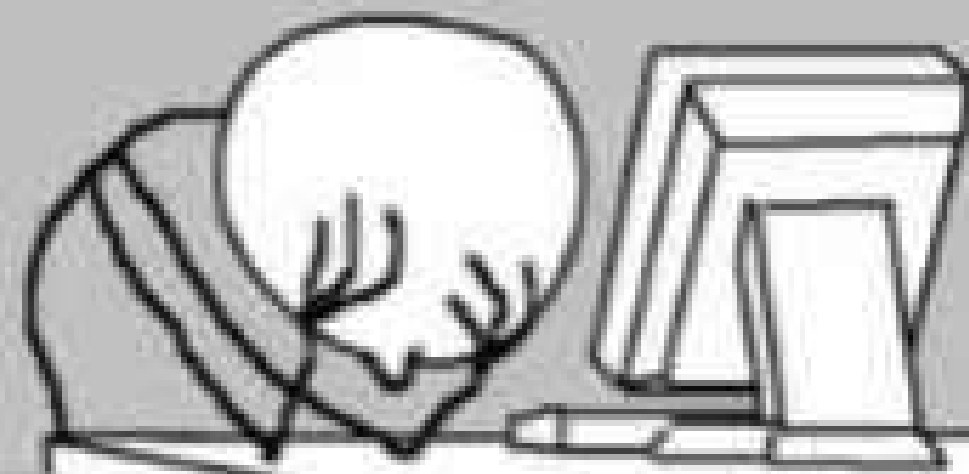
이 문제는 인간의 지능(추론)이 필요한가,
아니면 명확한 규칙(로직)이 필요한가?

Days before OpenAI

Developer coding
- 2 hours



Developer debugging
- 6 hours

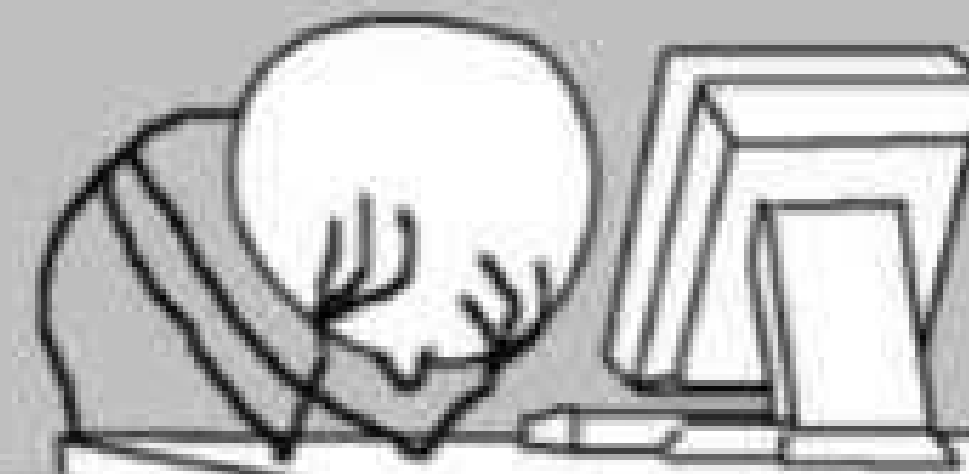


Days after OpenAI

ChatGPT generates
Codes - 5 min



Developer debugging
- 24 hours



Yes, AI 

- **방대한 비정형 데이터 처리**
- **초개인화된 사용자 경험**
- **창의적 초안 작성 및 브레인스토밍**
- **복잡한 추론과 계획**

No, AI 🚫

- **단순 계산 및 결정론적 로직**
- **실수가 치명적인 영역 (회계, 의료 등)**
- **비용 효율성이 떨어지는 단순 작업**
- **설명 가능성이 필수인 경우**

COMPLEXITY

Quickest Wins

high complexity low precision	high complexity high precision
low complexity low precision	low complexity high precision

Most Value

PRECISION

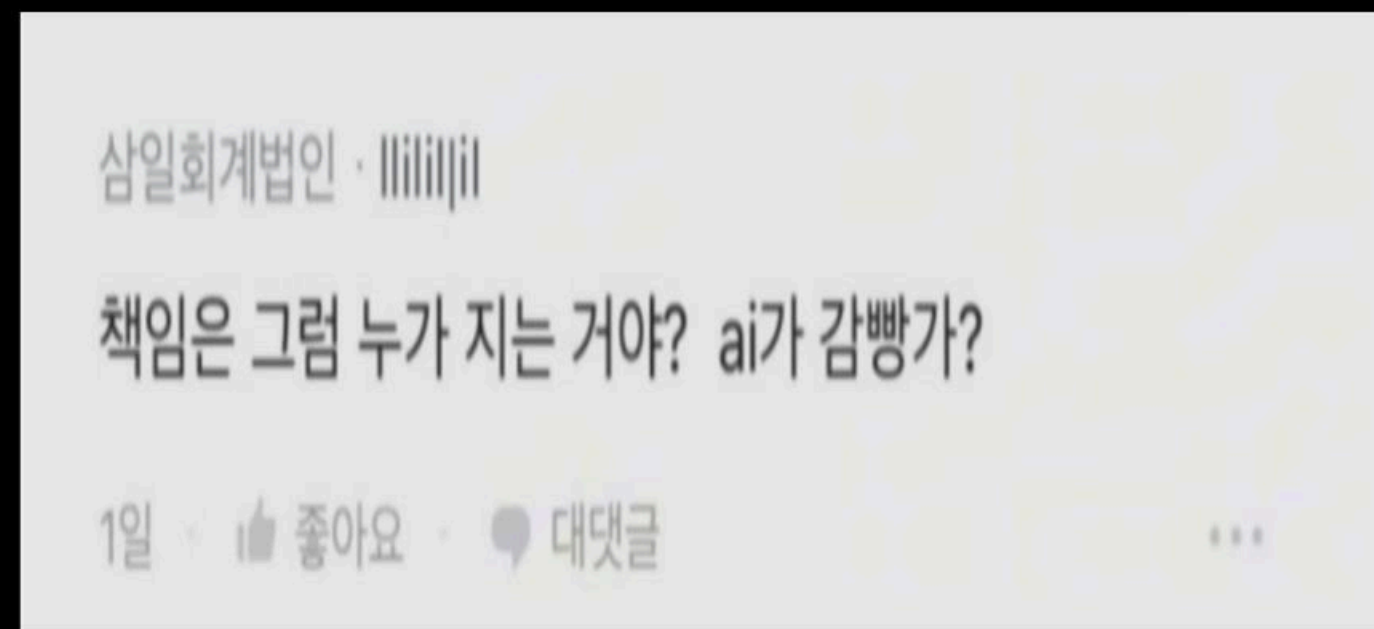
When to AI

구분	AI가 정답인 경우 (Use AI)	전통적 방식이 정답인 경우 (No AI)
데이터 형태	텍스트, 이미지, 음성 등 비정형 데이터	숫자, 데이터베이스 표 등 정형 데이터
정답의 성격	"괜찮은 수준"의 여러 답이 존재함	"단 하나의 정확한 정답"이 존재함
복잡도	변수가 너무 많아 규칙으로 정하기 힘들	명확한 비즈니스 로직(If-Then)이 존재함
사용자 니즈	"나만을 위한 맞춤형 결과"를 원함	"빠르고 일관된 결과"를 원함
실수 영향	약간의 실수는 교정 가능함 (저위험)	실수가 법적/경제적 재앙이 됨 (고위험)

AI 도입시 필수 질문

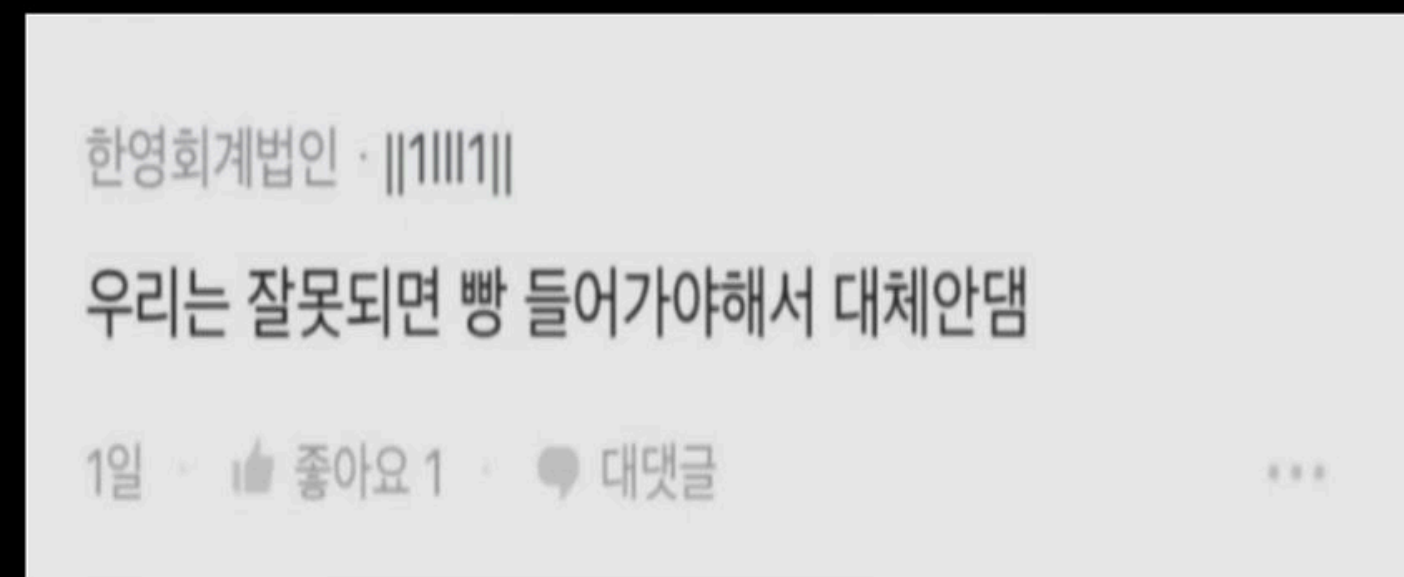
- AI로 어떤 의사결정을 바꿀 것인가?
- AI로 어떤 업무 흐름을 줄이거나 제거할 것인가?
- AI가 개입하면서 책임 구조는 어떻게 달라지는가?

회계사들이 AI를 두려워하지 않는 이유



"책임은 그럼 누가 지는 거야?"

회계사들이 AI를 두려워하지 않는 이유



"빵 가야 돼서 대체 안 됨"

회계사들이 AI를 두려워하지 않는 이유



회계사들은 감빵 누가 갈지

AI 도입시 필수 질문

- 어떤 시스템에 AI를 붙일 것인지?
- 기존 프로세스를 어디서 끊고 연결할 건지?
- 사람이 개입할 지점은 어디인지?
- 실패했을 때 되돌릴 수 있는지?

비즈니스 임팩트 평가

1. 생산성 향상의 정량적 공식 ($F \times T \times N$)

Total Value = Frequency x Time x # of Users

- 업무 빈도(F)
- 절감 시간(T)
- 대상 인원수(N)

비즈니스 임팩트 평가

2. 질적 임팩트: 병목 현상(Bottleneck) 해소

단순히 시간을 줄이는 걸 넘어, 그 업무가 지체됨으로써 발생하는 "기회비용" 고려하기

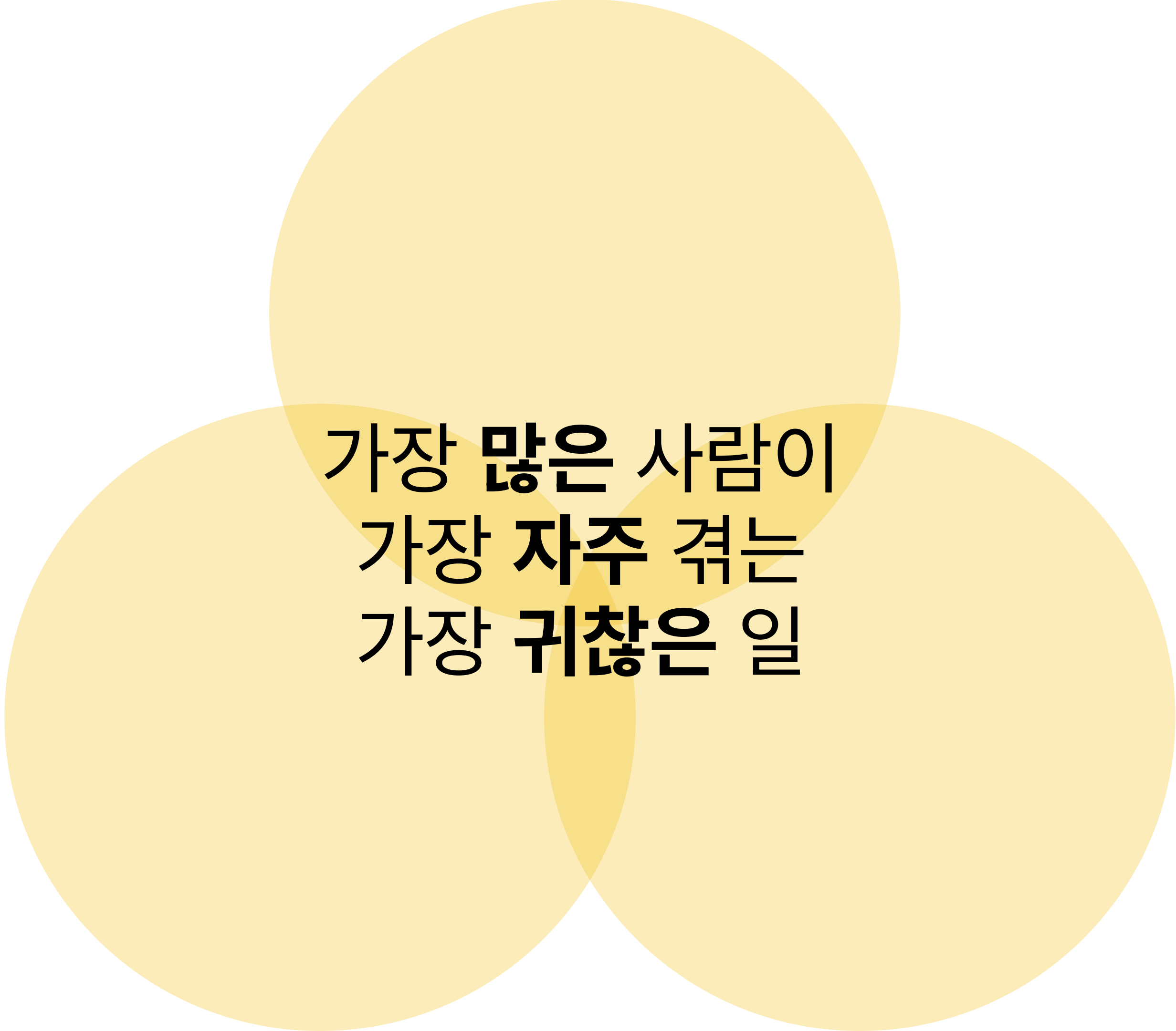
- 의사결정 속도
- 심리적 해방

비즈니스 임팩트 평가

3. 우선순위 매트릭트: 임팩트 vs 난이도

모든 문제를 한 번에 해결할 수 없으므로, '임팩트'를
축으로 하여 우선순위 정하기

- **High Impact, Low Effort**
- **High Impact, High Effort**
- **Low Impact, Low Effort**
- **Low Impact, High Effort**



가장 많은 사람이
가장 자주 겪는
가장 귀찮은 일

Product Market Fit

진짜 해결할 가치가 있는 문제 찾기

**“문제를 해결하는데 1시간이
주어진다면, 2/3 이상의 시간을
문제가 무엇인지 정의하는데 쓸 것이다.”**

Matthew Wakeman, “UX Research Practical
Techniques for Designing Better Products”

스타트업의 90%는 망한다.

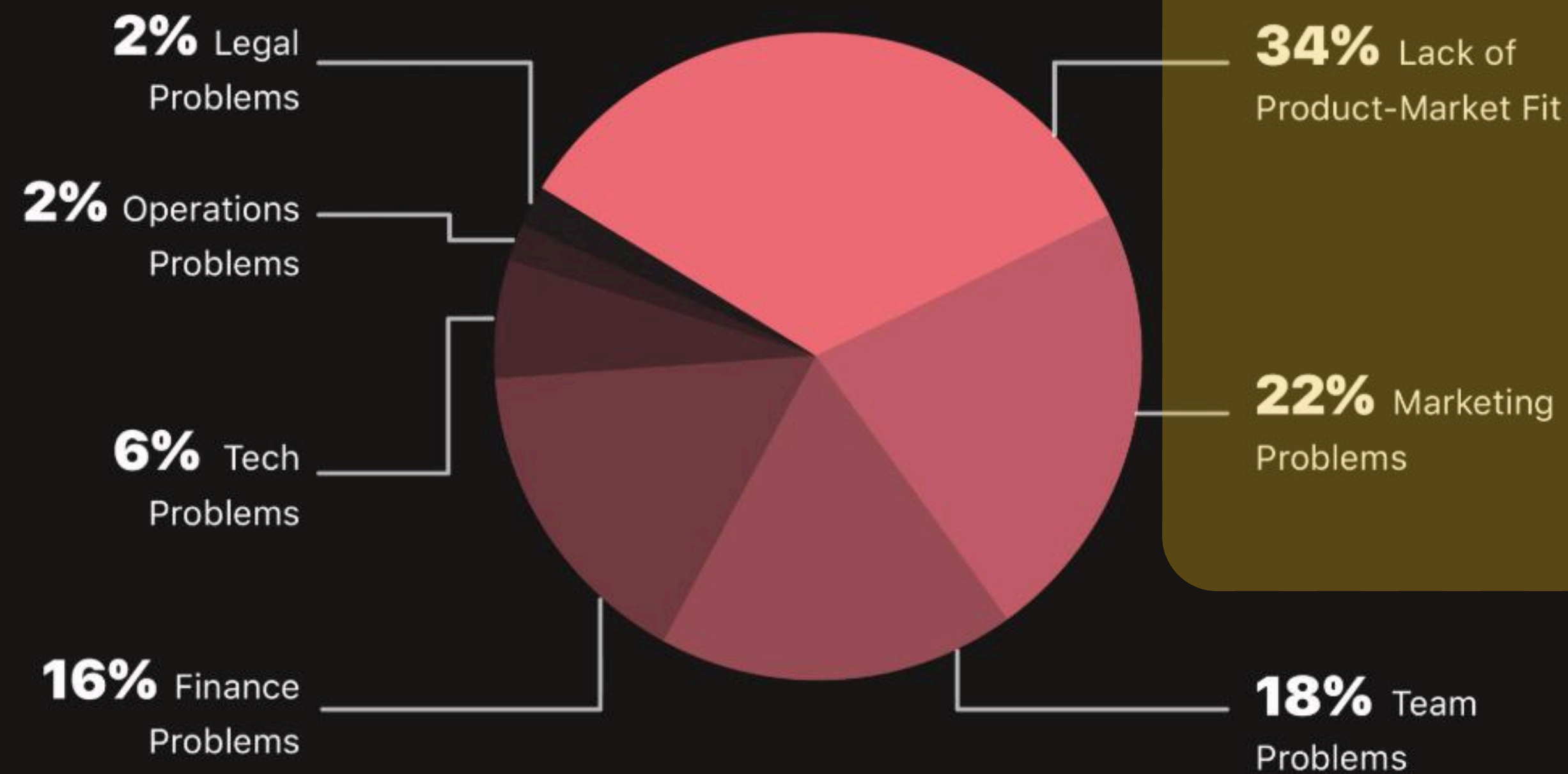
Failure Rate Of All Startups



More than **9 out of 10** startups fail.

대부분 쓸데없는 문제를 골라서 망한다.
(=아무도 안쓰는 제품을 만들어서 망함)

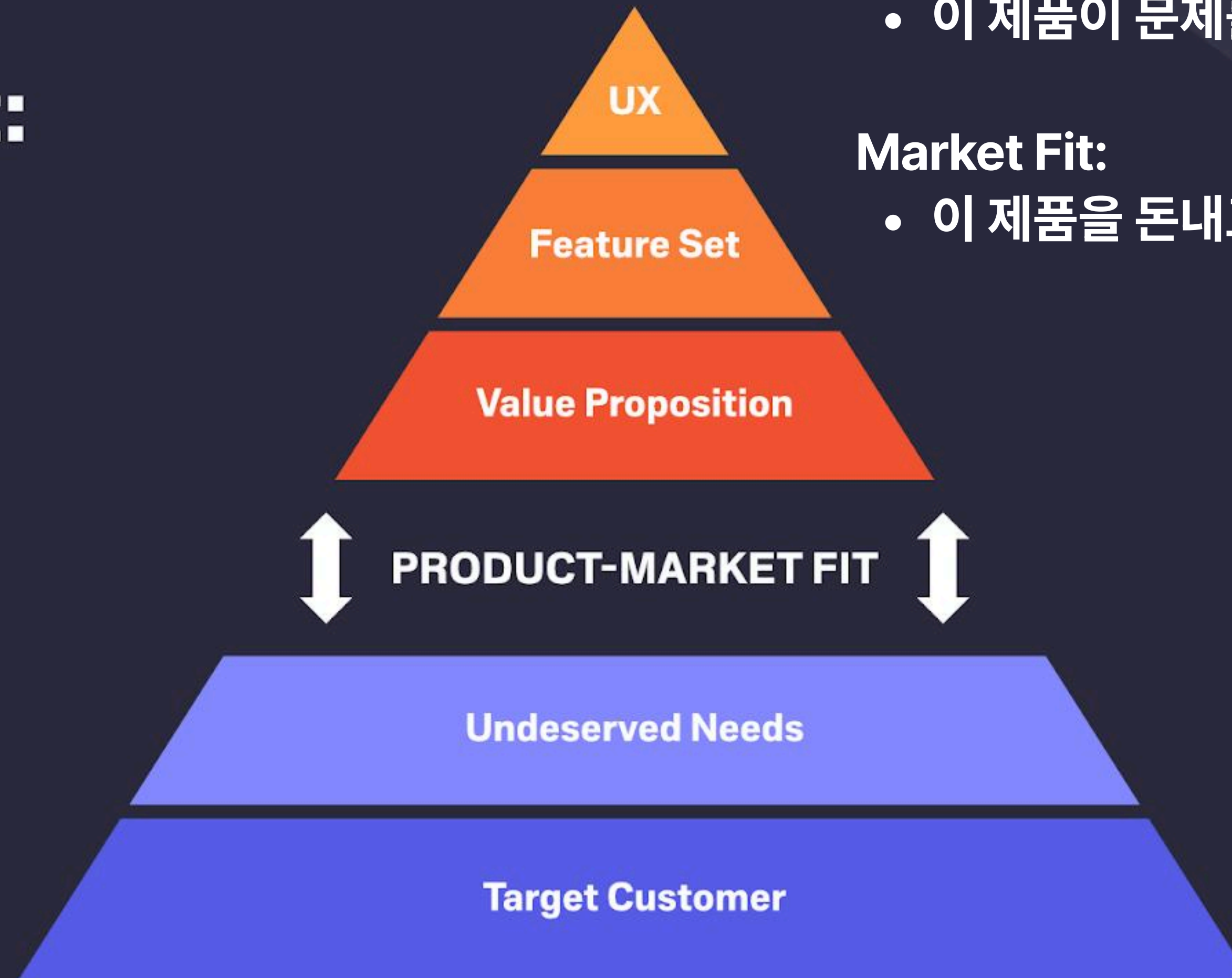
Common Reasons For Startup Failure



Information from 80+ failed startup interviews we've carried out.

그래서 필요한게 Product-Market-Fit (PMF)

Product:



Product Fit:

- 이 제품이 문제를 정말 해결해주나?

Market Fit:

- 이 제품을 돈내고 쓸 사람이 충분한가?

Market:



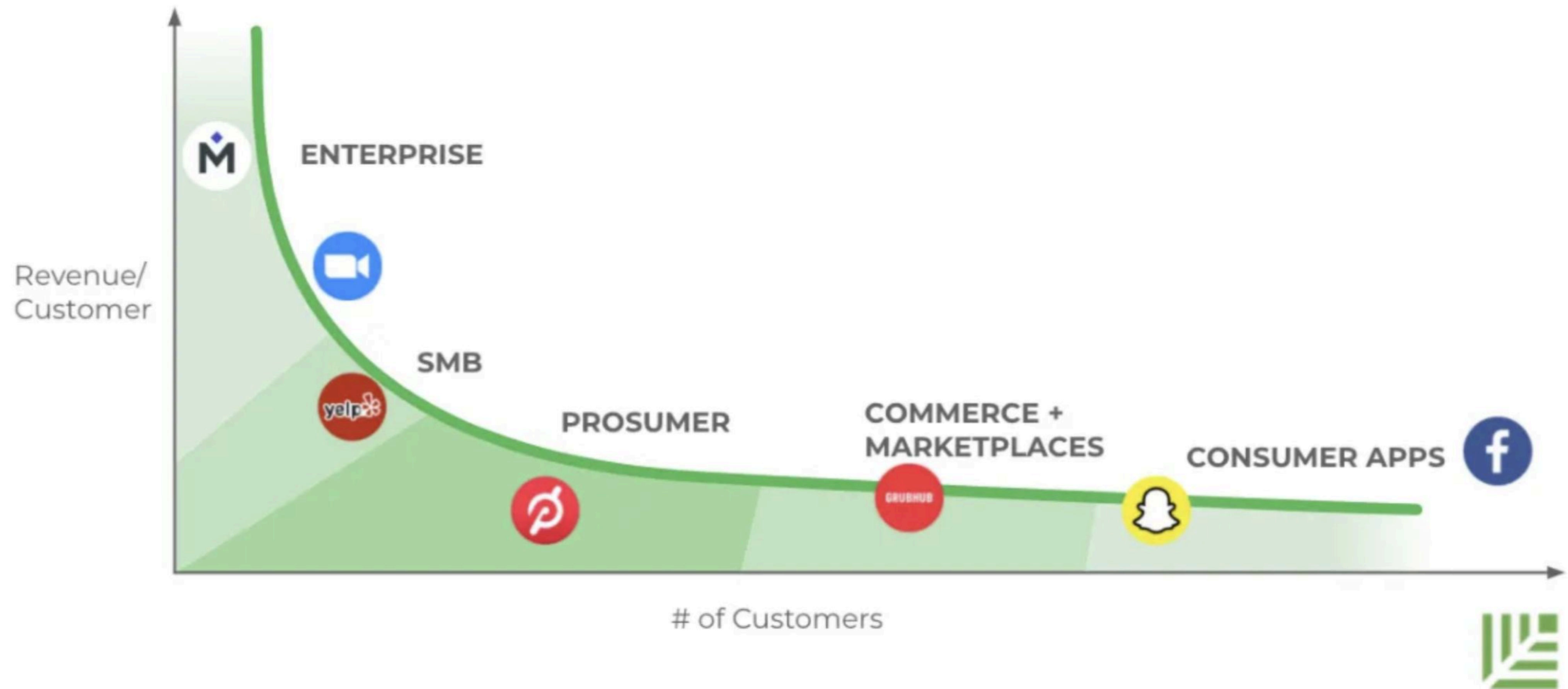
**“스타트업을 죽이는 가장
큰 요인은 시장의 부재다.”**

“The #1 company-killer
is lack of market.”

Marc Andreessen,
the general partner of Andreessen Horowitz / VC

When a great team meets a lousy market, market wins.
When a lousy team meets a great market, market wins.
When a great team meets a great market, something special happens.

훌륭한 팀이 가망없는 시장에 뛰어들면, 시장이 이긴다.
형편없는 팀이 훌륭한 시장에 뛰어들면, 시장이 이긴다.
훌륭한 팀이 훌륭한 시장을 만나면 뭔가 엄청난 일이 생긴다.

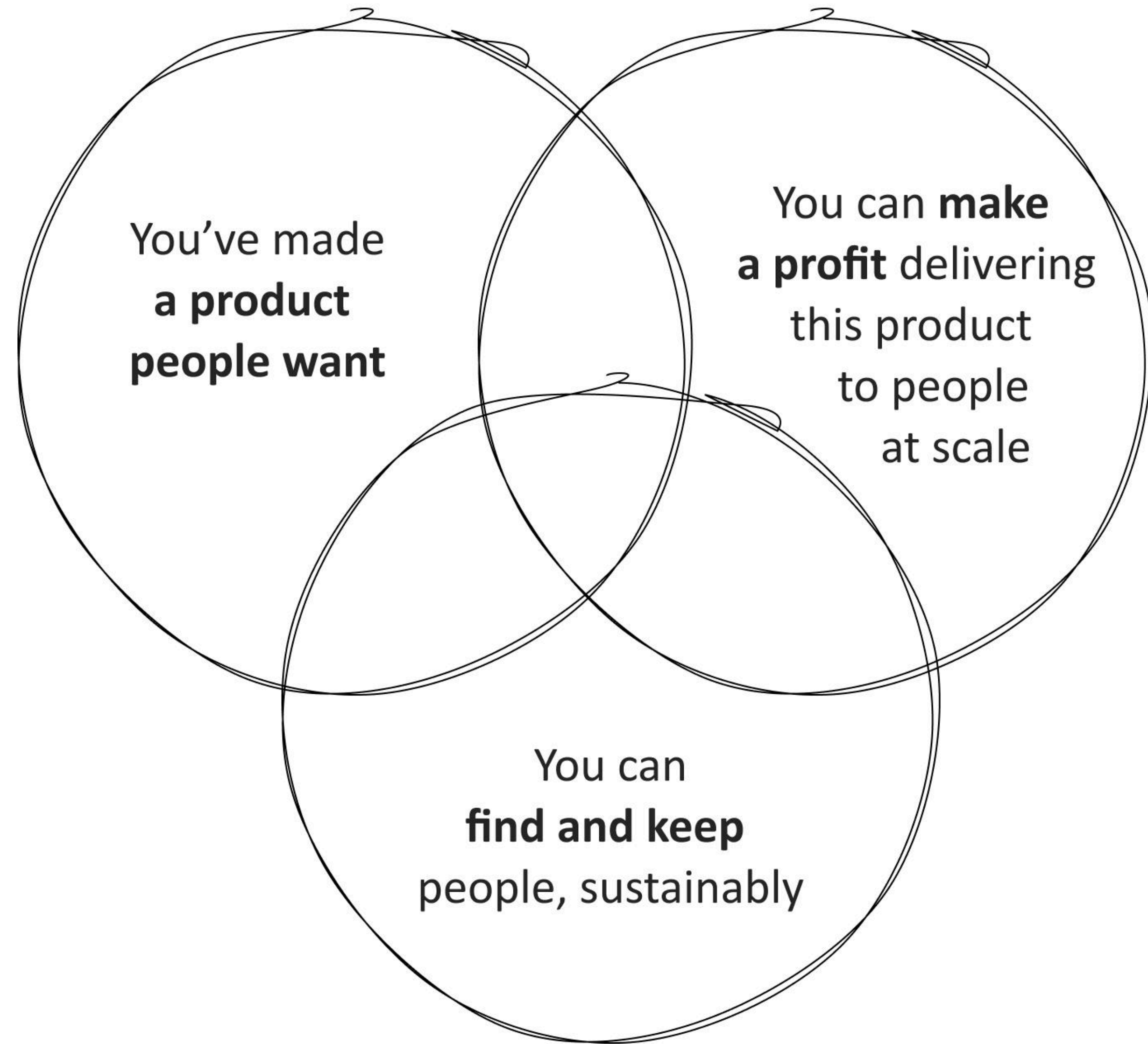


<https://medium.com/sequoia-capital/the-market-curve-44097b626f6d>

시장 사이즈 = 고객 수 * 고객당 수익

Market Size = # of Customers * Revenue per Customer

When do I have
**True Product-
Market Fit?**

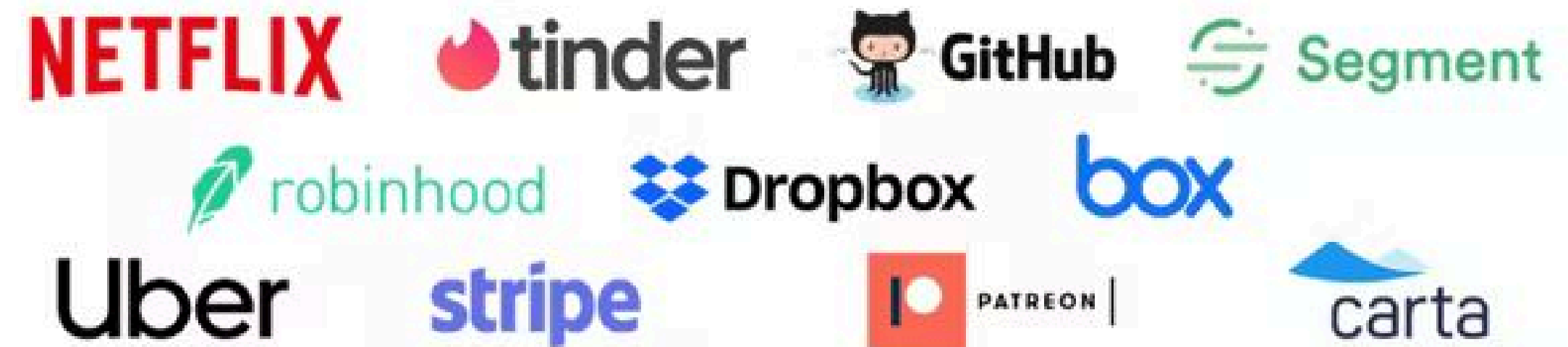


PMF 초기 신호

1. 유료 고객 확보
2. 강력한 감정적 반응
3. 자발적인 고객 유입
4. 지속적인 사용

Three signs you've found Product-Market Fit

 **Sudden and significant pull**
from the market



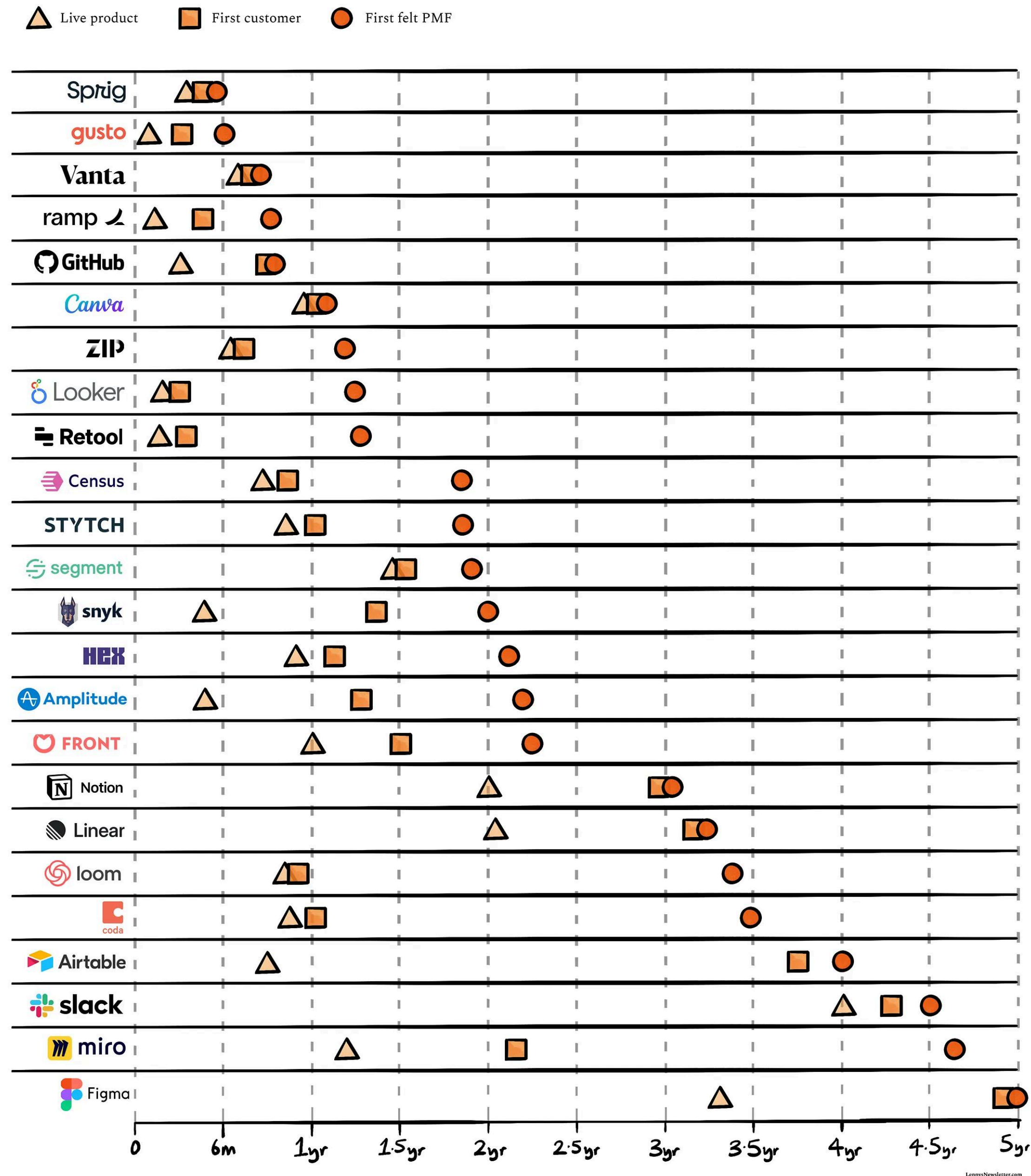
 **Gradual but compounding pull**
from the market



 **Hitting a milestone**
that proves it's working

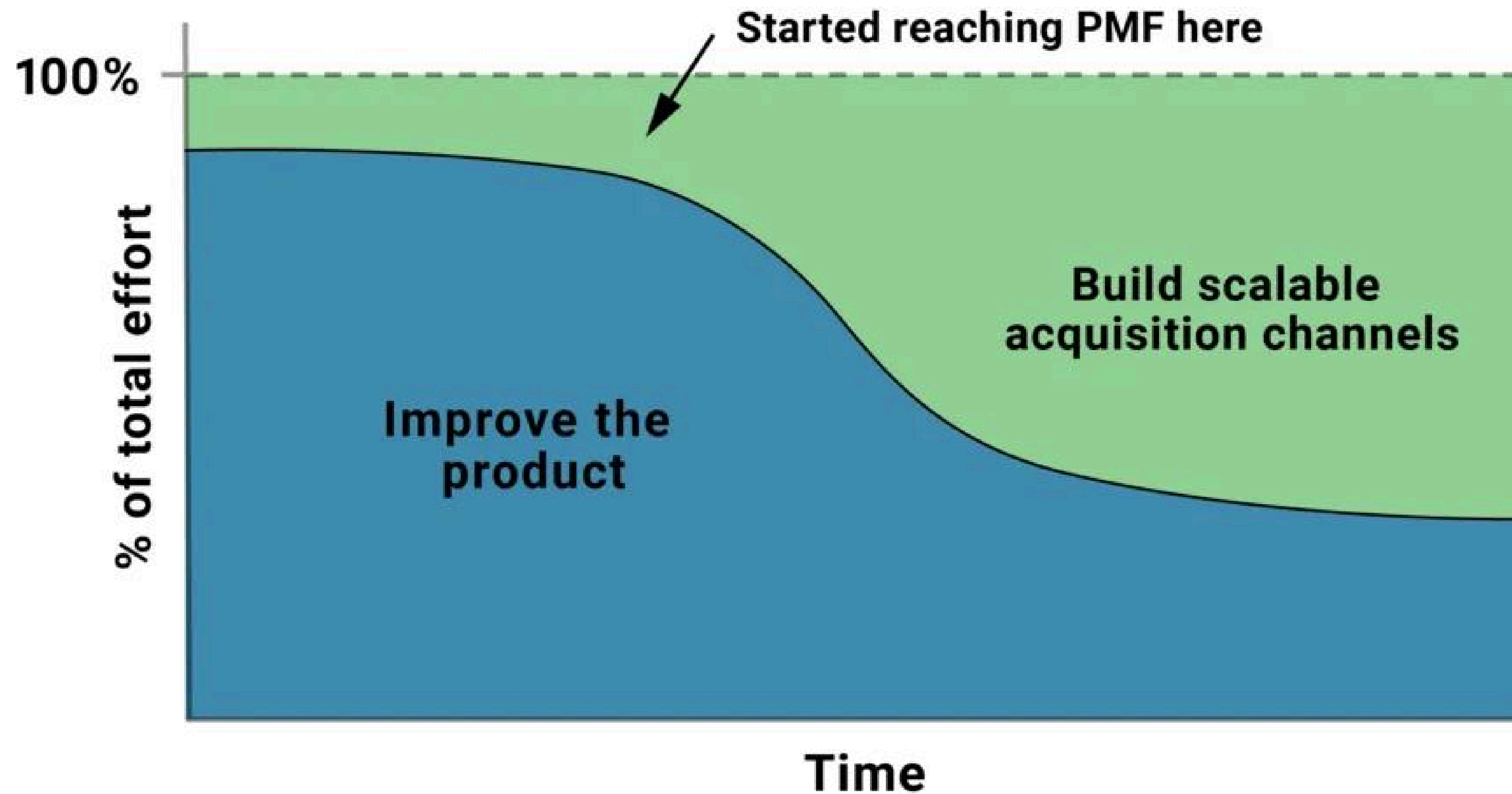


How long it took from idea to product-market fit



B2B 시장에서 PMF를 찾기까지 걸리는 시간은 최소 2년 ~ 평균 4년이 소요

- Github: 2년 정도
- Notion: 3년 정도
- Slack: 4년 정도
- Figma: 5년 정도

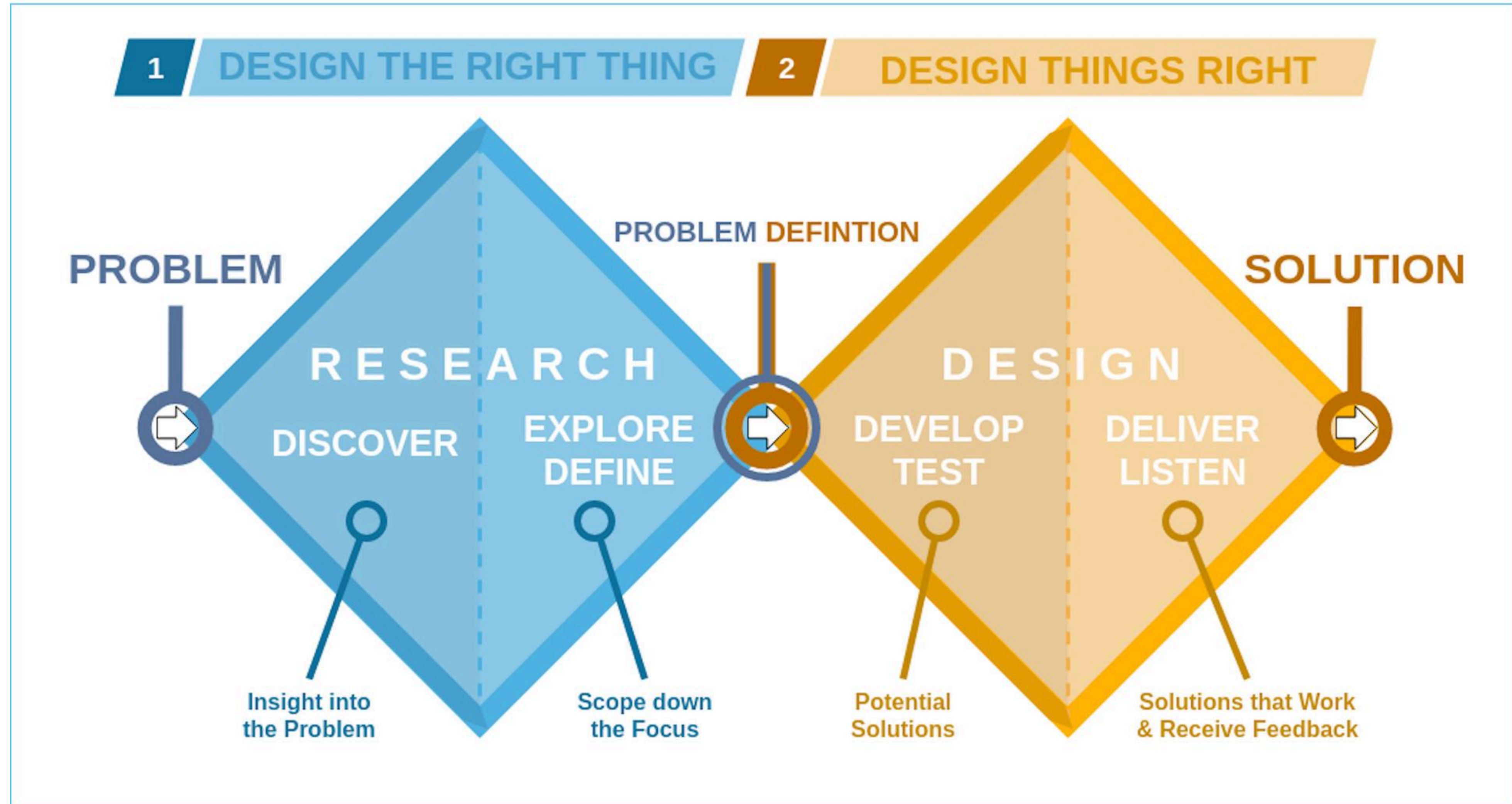


growthengblog.com

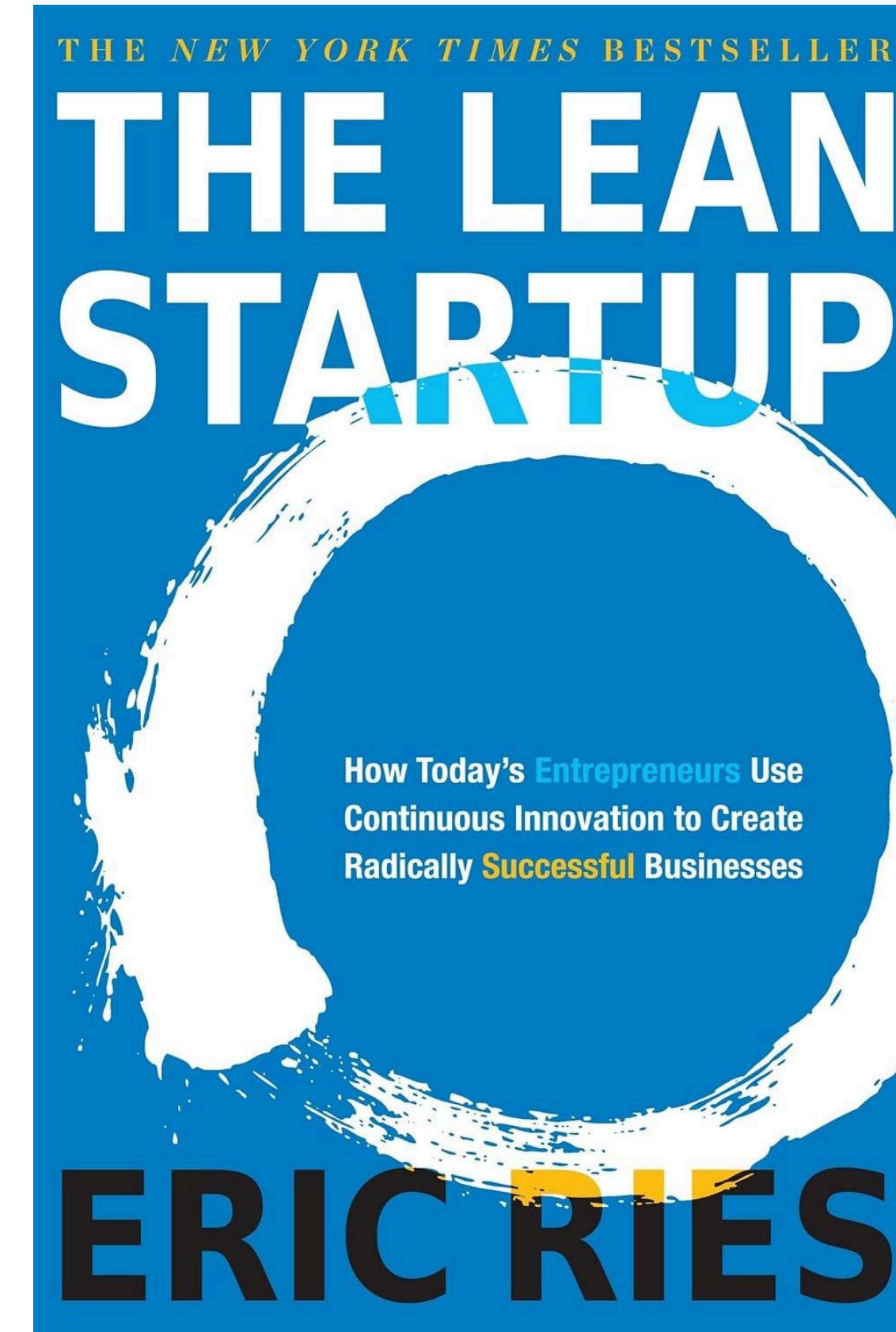
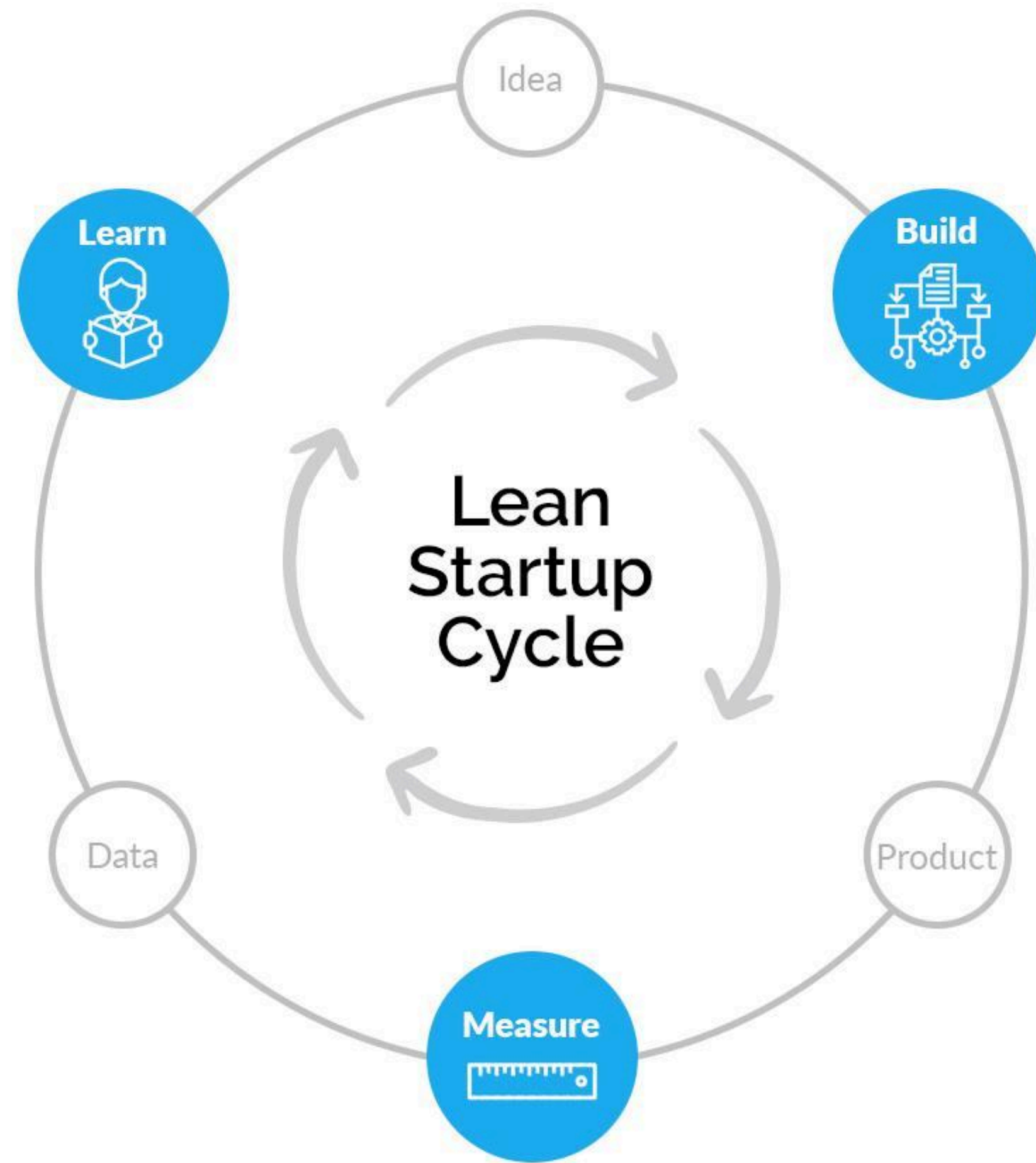
PMF는 한 번 찾고
끝나는게 아니라,
끊임없이 시간에 따라
다듬어 나가는 것

Pre-PMF vs
Post-PMF

Double Diamond Process

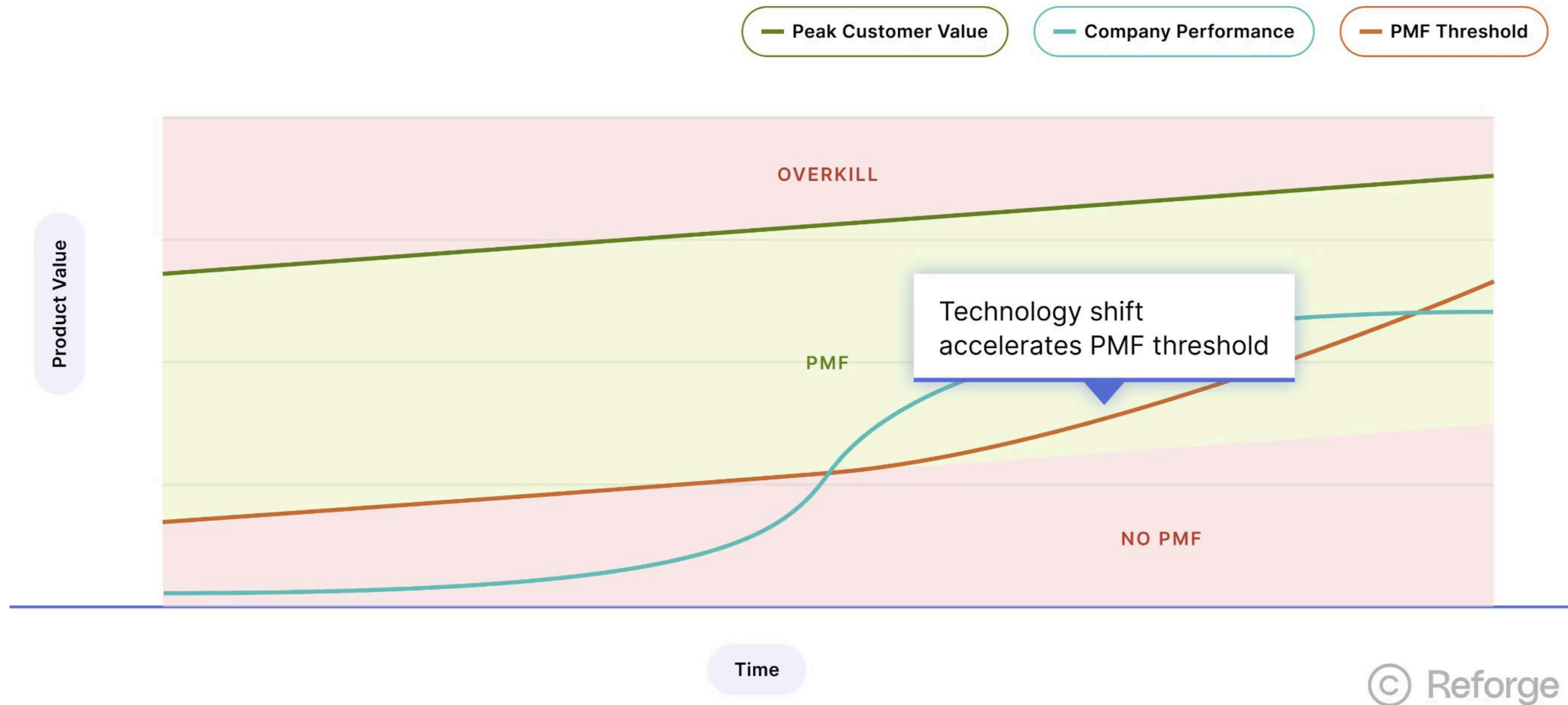


린 스타트업 (Lean Startup)

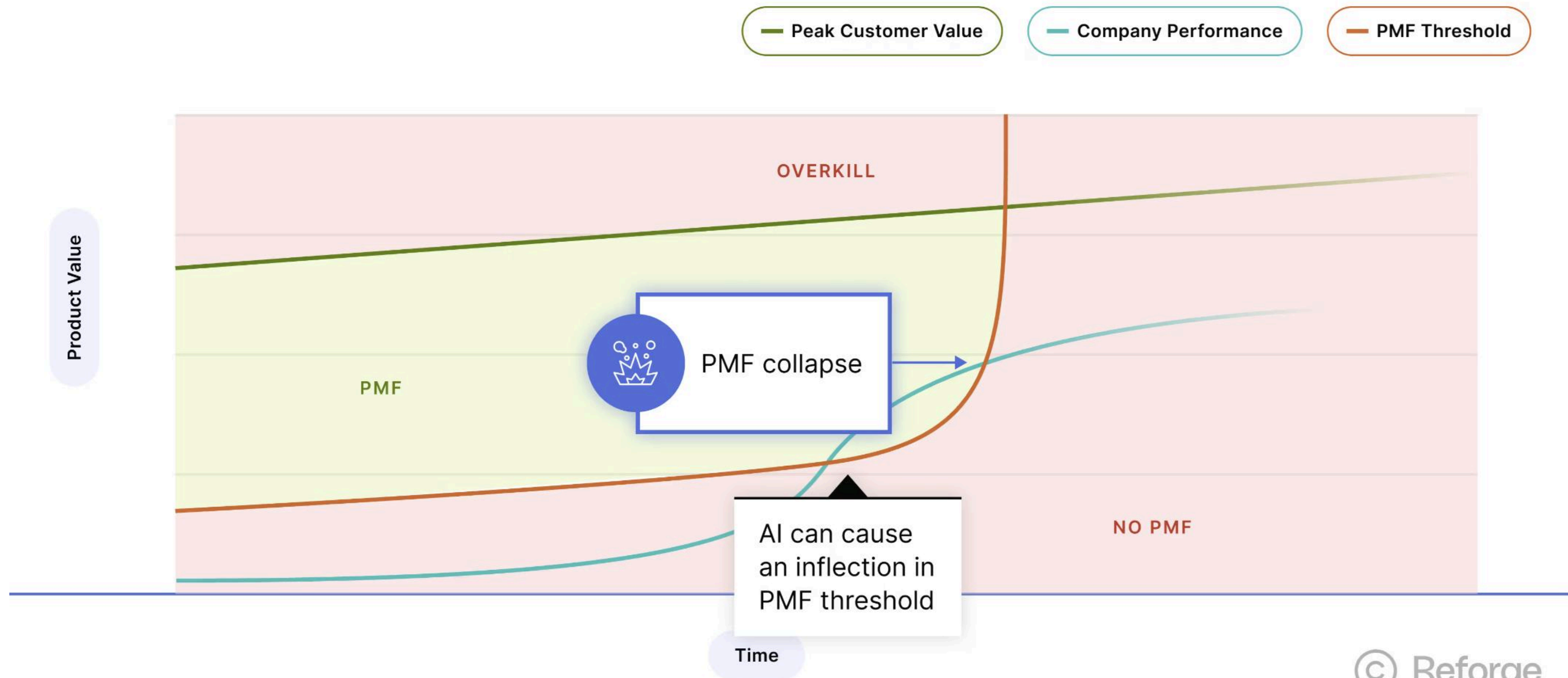


PMF의 붕괴(?)

PMF Treadmill



Product Market Fit Collapse



Product M

Product Value



Thinking
PMF means
you're safe



Realizing
PMF moves
every month
thanks to AI



AI PMF = 움직이는 과녁

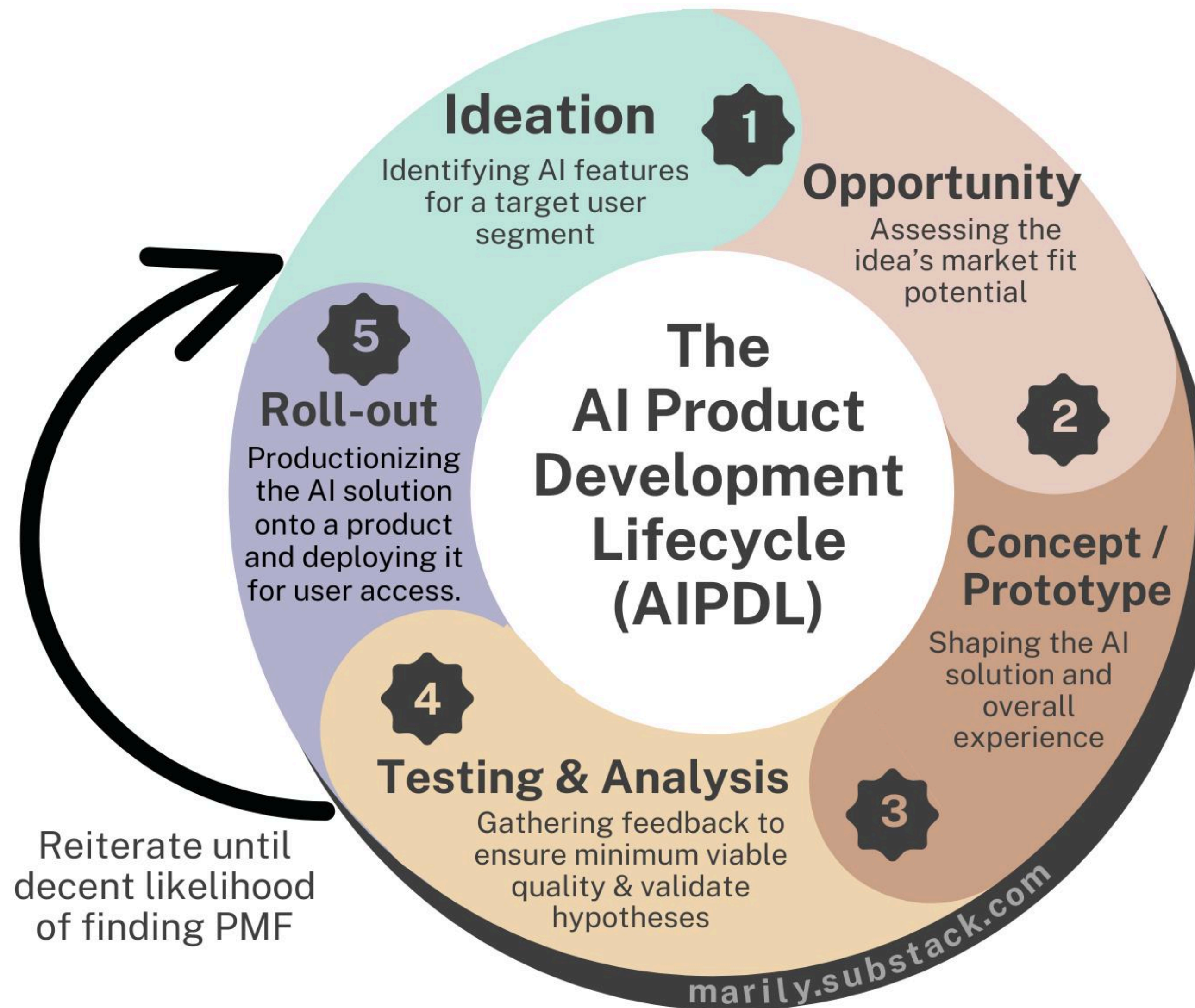
PMF is a moving target
Measure. Track. Adjust. Repeat.



기존 서비스로는 불가능했던 획기적인 경험을 AI로 제공하는 것

“In today’s environment, PMF increasingly involves not just solving customer problems, but solving them in ways that *leverage AI’s unique capabilities to create experiences that were previously impossible.*”

AI Product Lifecycle



AI Development Life Cycle Stages

01

Problem Identification And
Requirement Analysis

02

Data Collection and
Preparation

03

Model Selection and
Training

04

Model Development

05

Model Evaluation and
Validation

06

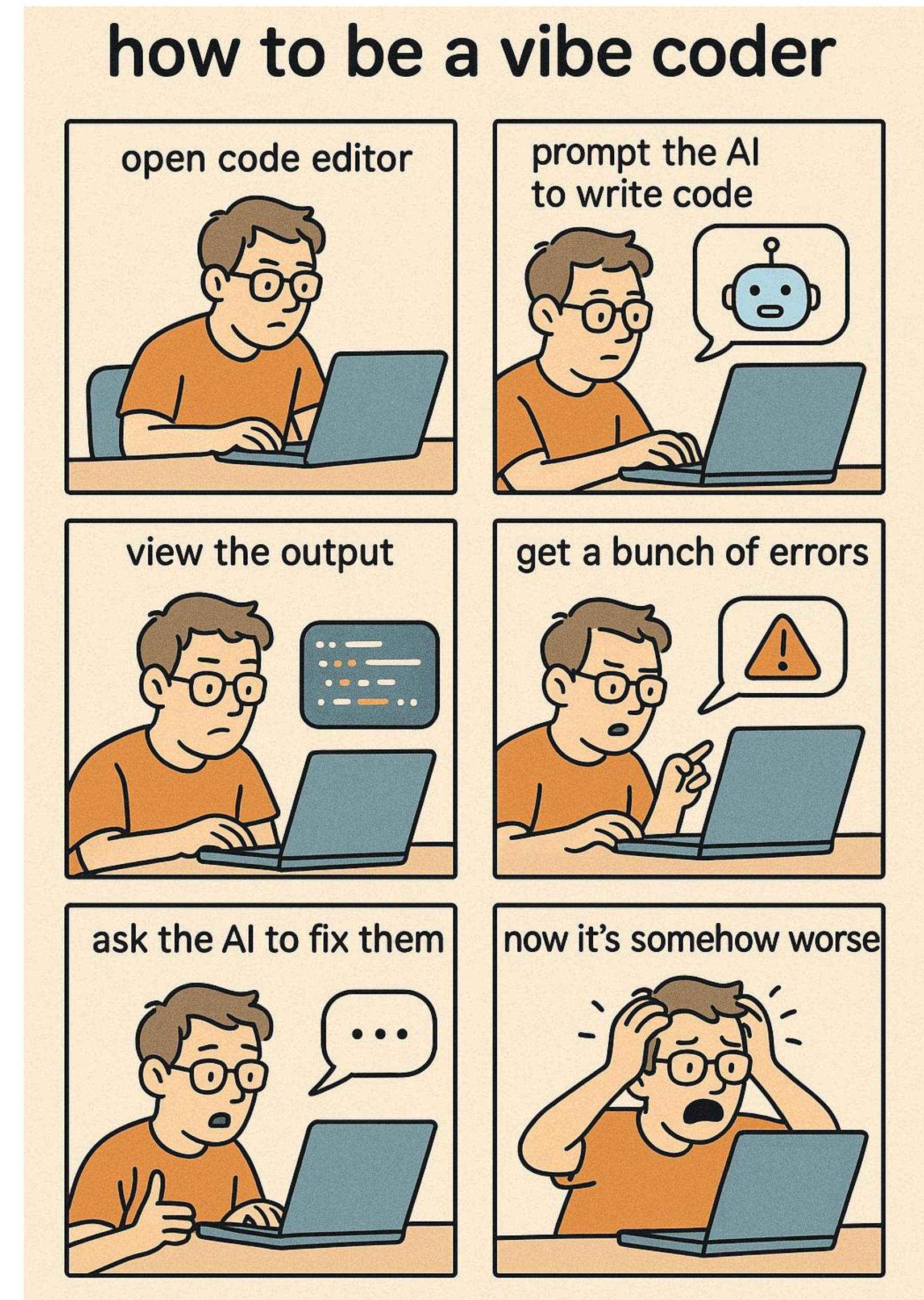
Deployment and
Integration

07

Monitoring and
Maintenance

현실적인 개발 사이클

아이디어 → 대충 만들어봄 → 돌려봄
→ 이상하게 동작 → 프롬프트 수정
→ 다시 돌림 → 좀 나아졌는데 다른 문제 발견
→ 도구 권한 조정 → 다시 돌림
→ (무한 반복) → 어느 정도 안정화



조금 더 고급 버전

근데 프롬프팅만은 아닙니다

테스트하면서 조정하는 건 프롬프트만이 아니에요:

조정 대상	예시
프롬프트	지시가 모호했네 → 더 구체적으로 수정
도구 권한	Write 줬더니 막 수정하네 → Read만 허용
스킬 조합	이 지식이 빠졌네 → 스킬 추가
모델 선택	sonnet으로 부족하네 → opus로 변경
maxTurns	무한 루프 돌아 → 턴 제한
완료 조건	중간에 끝내버리네 → 조건 명확히
구조 자체	에이전트 하나론 안 되네 → 여러 개로 분리

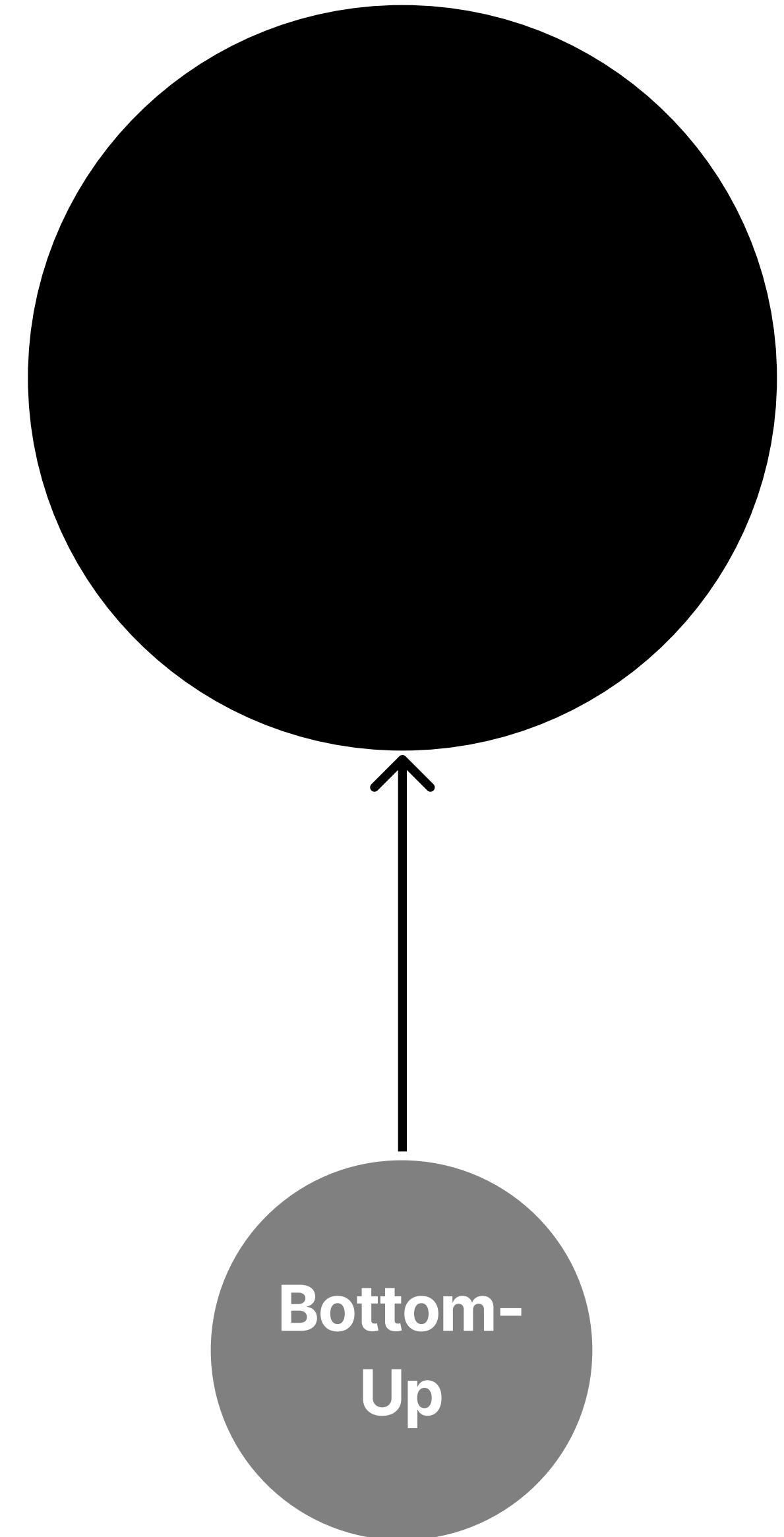
시장 찾기

Market Fit

시장은 어떻게 찾냐(1)

Bottom-up Method

- 내가 일상에서 겪고 있는 가장 큰 문제는?
- 다른 사람들도 같은 문제를 겪고 있는가?
- 시장에서 아직 해결되지 않은 문제인가?



1. 과거의 경험에서 찾기 (Past Pain)

- 이전 직장에서 가장 가치 있었던 솔루션은 무엇이었는가?
- 과거에 "돈을 내고서라도 쓰고 싶다"고 생각했던 툴이 있었는가?
- 여러 회사에서 반복적으로 만들었던 내부용 제품은 무엇인가?

2. 현재 시장에서 탐색하고 검증하기 (Ponder and Probe)

- 새롭게 떠오르지만 아직 서비스가 부족한 트렌드는 무엇인가? (예: 보안, 협업)
- 충분히 활용되지 않고 있는 혁신 기술은 무엇인가? (예: 데이터 확장성)
- 잠재 고객 인터뷰를 통해 문제의 중요도와 시장 미충족 수준을 파악

3. 현재의 기회에서 발견하기 (Present Pull)

- 내가 만든 것 중 가장 고객의 반응(Pull)이 좋은 기능은 무엇인가?
- 지금 당장 내 스타트업이 겪고 있는 문제 중 해결하고 싶은 것은 무엇인가?
- 계속 미뤄왔던 사이드 프로젝트 아이템이 있는가?



**“경쟁은 멍청이들이나
하는 짓이다.”**

“Competition is
for losers.”

Peter Thiel,
Co-founder of PayPal, Palantir,
and the Founders Fund

**“성공한 실리콘밸리 기업의 대부분은
작은 시장에서 출발하여 점차 확장했다.”**

Peter Thiel,
Co-founder of PayPal, Palantir,
and the Founders Fund

facebook

하버드 학생 대상으로 시작

amazon

종이책 판매에서 시작

PayPal

eBay 파워셀러 대상 시작



Niche is Powerful

Twitch Co-Founder
저스틴 칸 (Justin Kan)

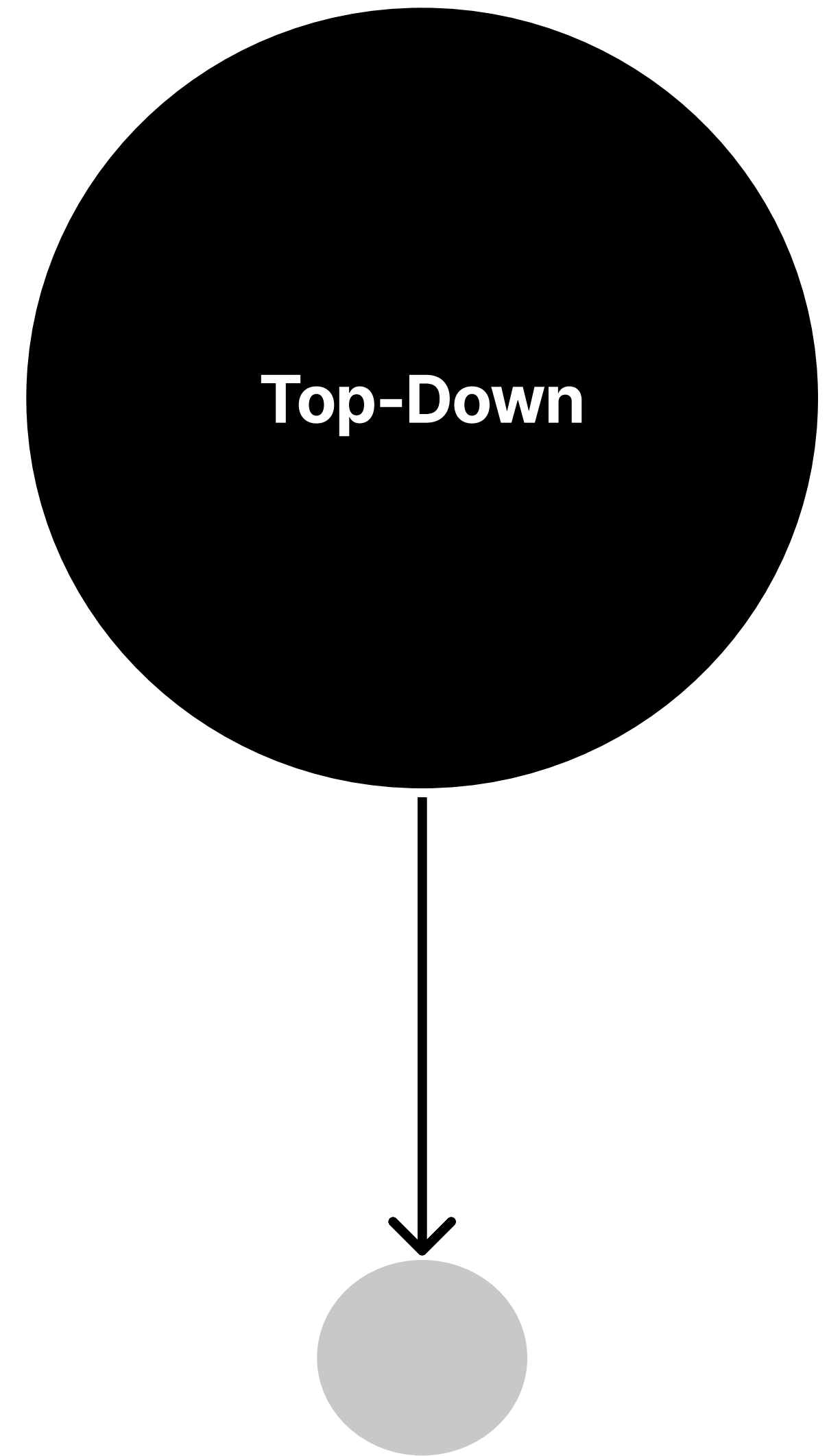
대학생 때 트위치를 만들기
시작해 아마존에 1조원에 매각

**초기에 일상 Live streaming
에서 Game live streaming
으로 니치다운해서 성공함**

시장은 어떻게 찾냐?(2)

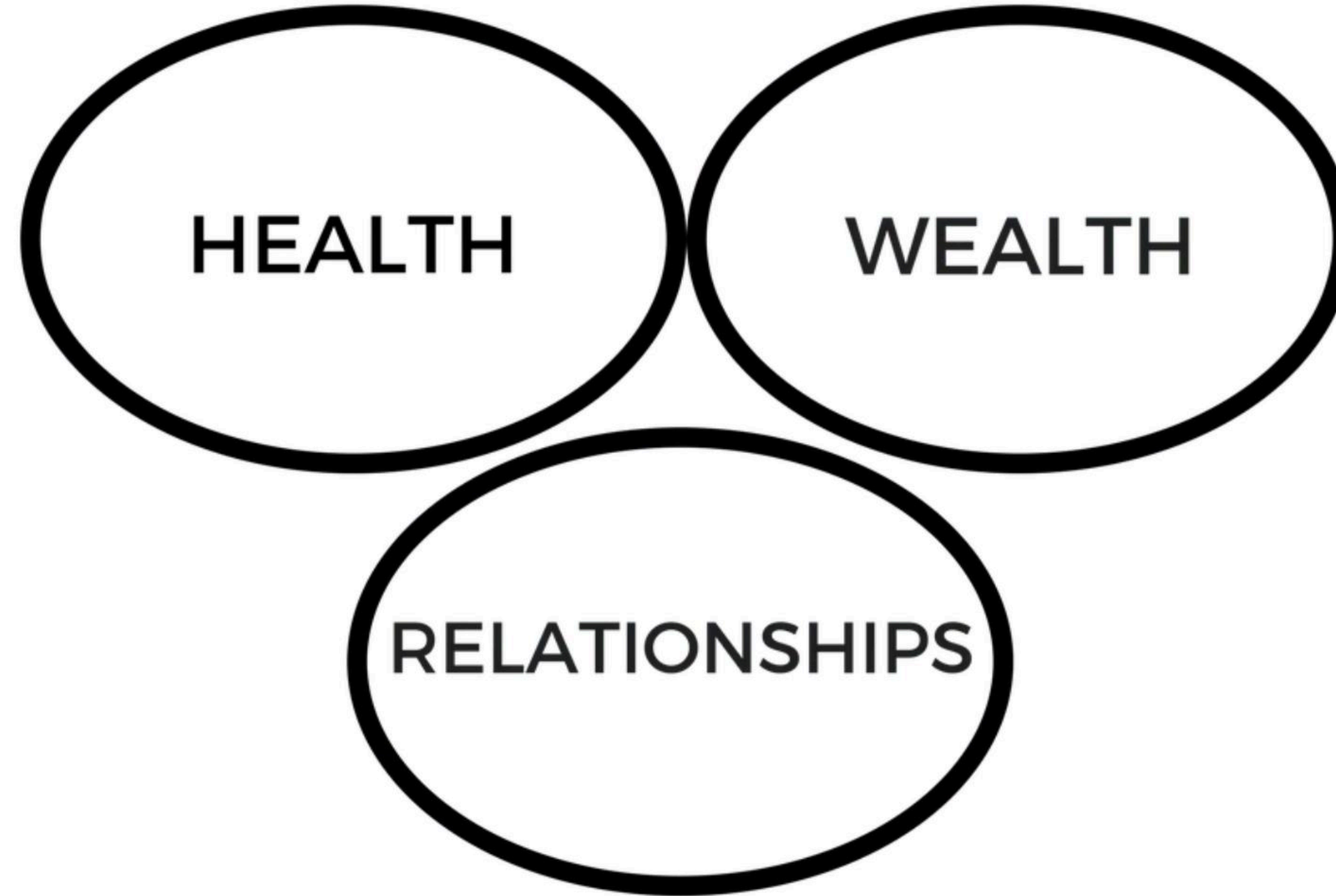
Top-Down Method

- 세상의 큰 흐름 속에서 새로운 문제 찾기
- 요즘 기술 트렌드는? 사회/정책적 변화는?



돈이 되는 3대 시장

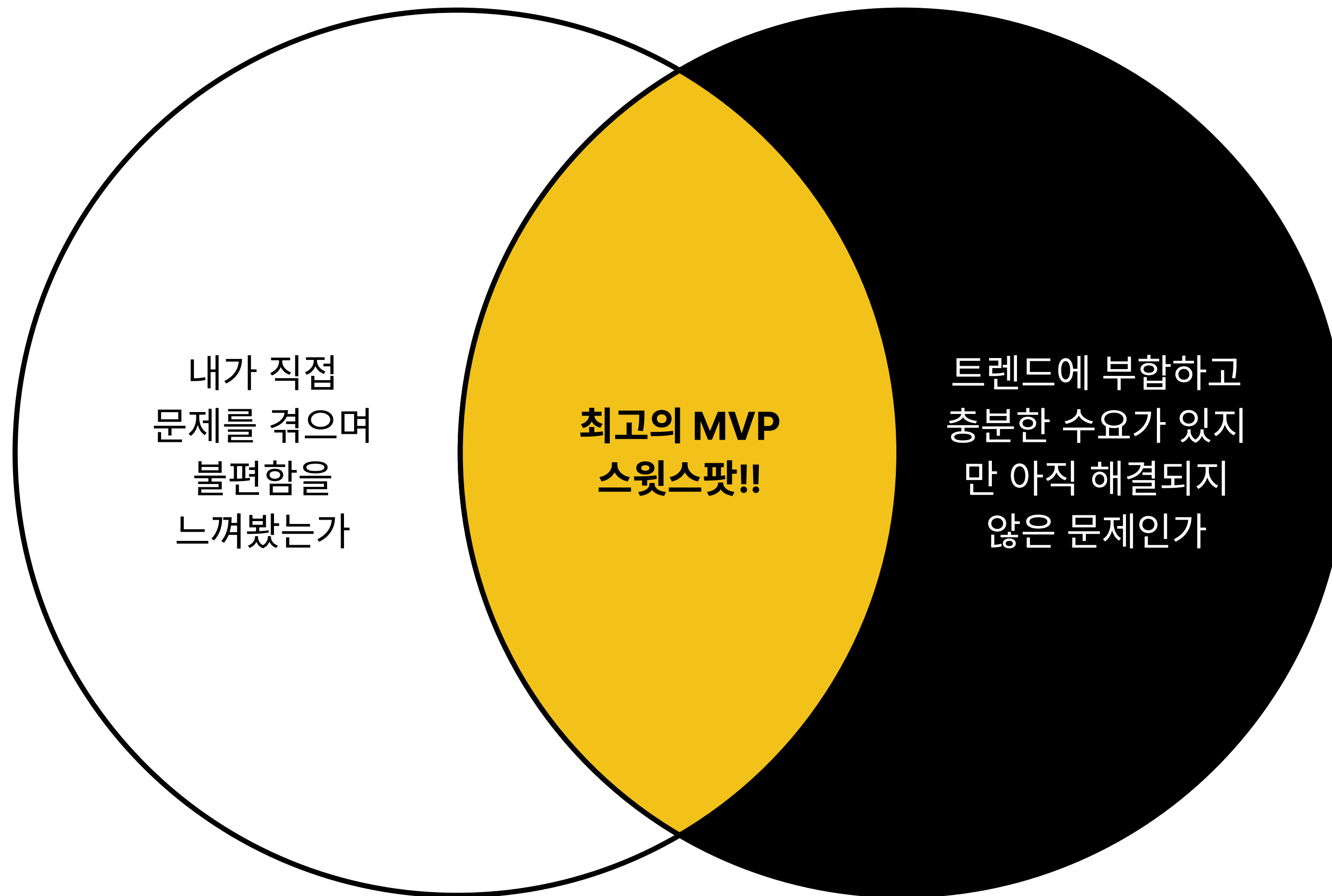
THE 3 MARKETS



요즘 뜨는 트렌드는 어디서 찾나?

- 각종 바이럴 SNS 콘텐츠
- 뉴스, 뉴스레터, 아티클
- 길거리 사람들, 광고 관찰
- 유튜브/인스타 댓글
- 온라인 커뮤니티, 단톡방
- 네이버, 구글 트렌드

성공할 확률이 가장 높은 시장

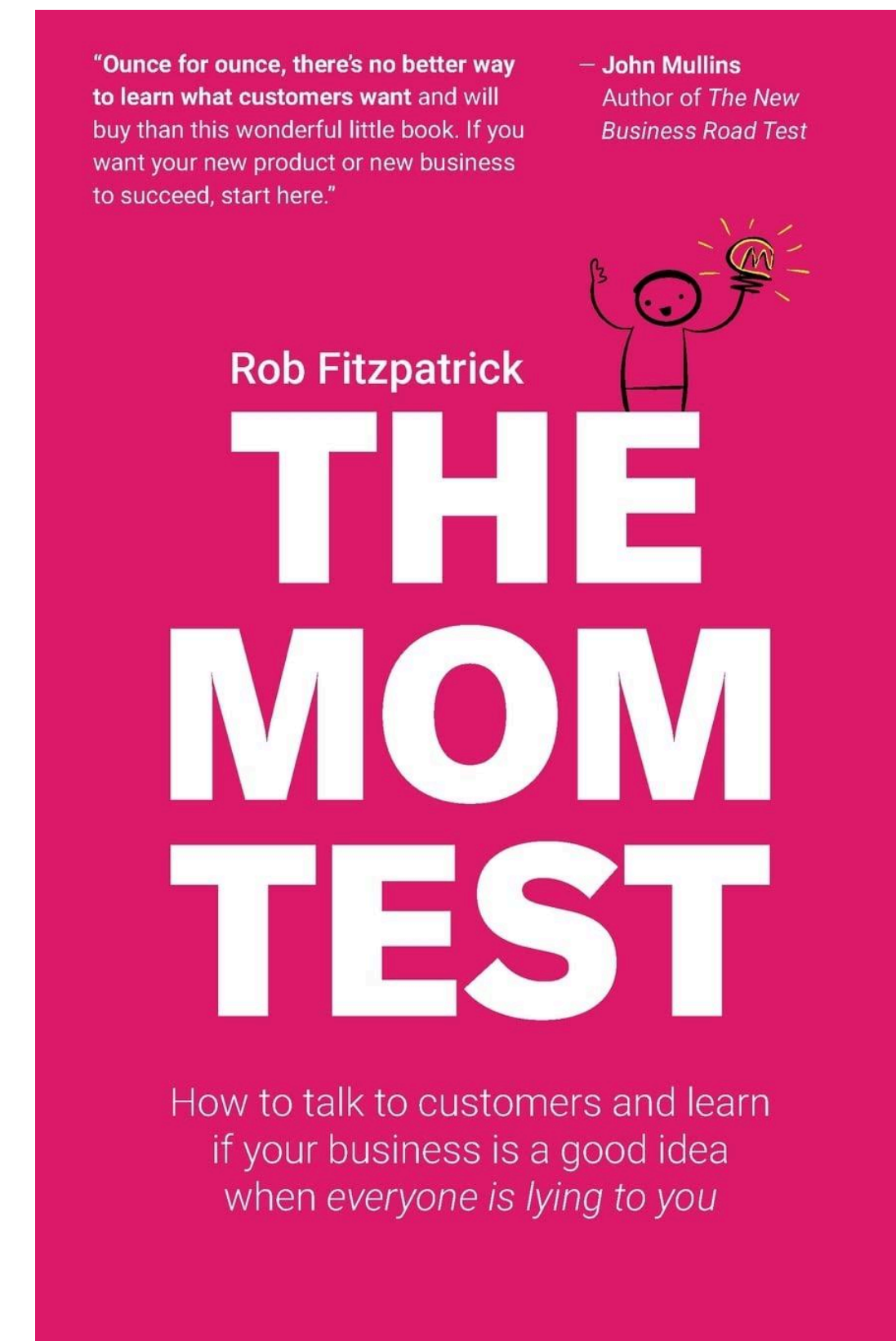


문제 찾기

Product Fit

뽕족한 페인포인트 찾기는 아직 AI가 정량화 하기 전의 영역!

- 고객들과 직접 교류하기!
- 내가 남보다 잘 아는거 하기!



당신의 아이디어에 대해 이야기하지 말고
고객의 삶과 문제에 대해 이야기하라.

The Mom Test (고객 인터뷰 방법론)

주제

❌ 나쁜 질문 (Bad Questions)

✅ 좋은 질문 (Good Questions)

아이디어
검증

"제가 지금 이러이러한 앱을 만들려고 하는데, 어떠세요?"

"최근에 OO 작업을 할 때 가장 불편하거나 시간이 오래 걸렸던 점은 무엇인가요?"

구매 의사

"이런 제품이 10만 원에 나온다면 구매하시겠어요?"

"그 문제를 해결하기 위해 이전에 돈을 써보신 적이 있나요? 있다면 얼마까지 써보셨어요?"

기능의 필요성

"저희 앱에 자동 리포트 기능이 있으면 유용할까요?"

"매달 리포트 작업에 보통 시간을 얼마나 쓰세요? 그 과정에서 가장 번거로운 부분은 무엇인가요?"

문제의 심각성

"보통 이 문제 때문에 많이 불편하신가요?"

"마지막으로 그 문제 때문에 화가 났거나 크게 불편했던 경험에 대해 자세히 말씀해주실 수 있나요?"

일반적인
의견

"당신에게 가장 중요한 것은 무엇인가요?"

"최근 일주일 동안 OO 업무에 사용한 시간 순서대로 우선순위를 매겨주실 수 있나요?"

고객의 삶 속에 들어가기

 Kurly Tech Blog

컬리로그팀이 직접 고객을 만나고 리서치를 진행한 이야기: 컬리 푸드 페스타에서

프로덕트 조직의 엔지니어가 고객을 직접 만나고 인터뷰한 이유



김영민 게시 날짜: 2023.08.28.



고객이 가지고 있는 진짜 문제를 찾고 해결하기 위해 프로덕트를 개발하는 팀이 직접 유저를 만나고 대화했습니다.

사무실 밖으로 나가서 우리는 무엇을 얻을 수 있었을까요?

사업 아이디어 노다지 캐는 법

X, Threads, Reddit에서 만들고자 하는
제품 관련 댓글 긁어와 AI로 VOC 분석

제품 관련 유튜브 & 인스타 콘텐츠 댓글,
경쟁사 서비스 리뷰 긁어서 AI로 VOC 분석

아이디어 브레인스토밍 프롬프트

- ▶ Market Idea Expander
- ▶ Google Query Reddit
- ▶ Pain Point Extractor
- ▶ Market Gap Generator
- ▶ Landing Page Prompt
- ▶ Transition Guardian

AI로 아이디어 발굴하고
뽕족한 제품 만드는 프롬프트

https://www.notion.so/junse/26be11b2513f809aba5ff7132b99f5f3?source=copy_link

왜?



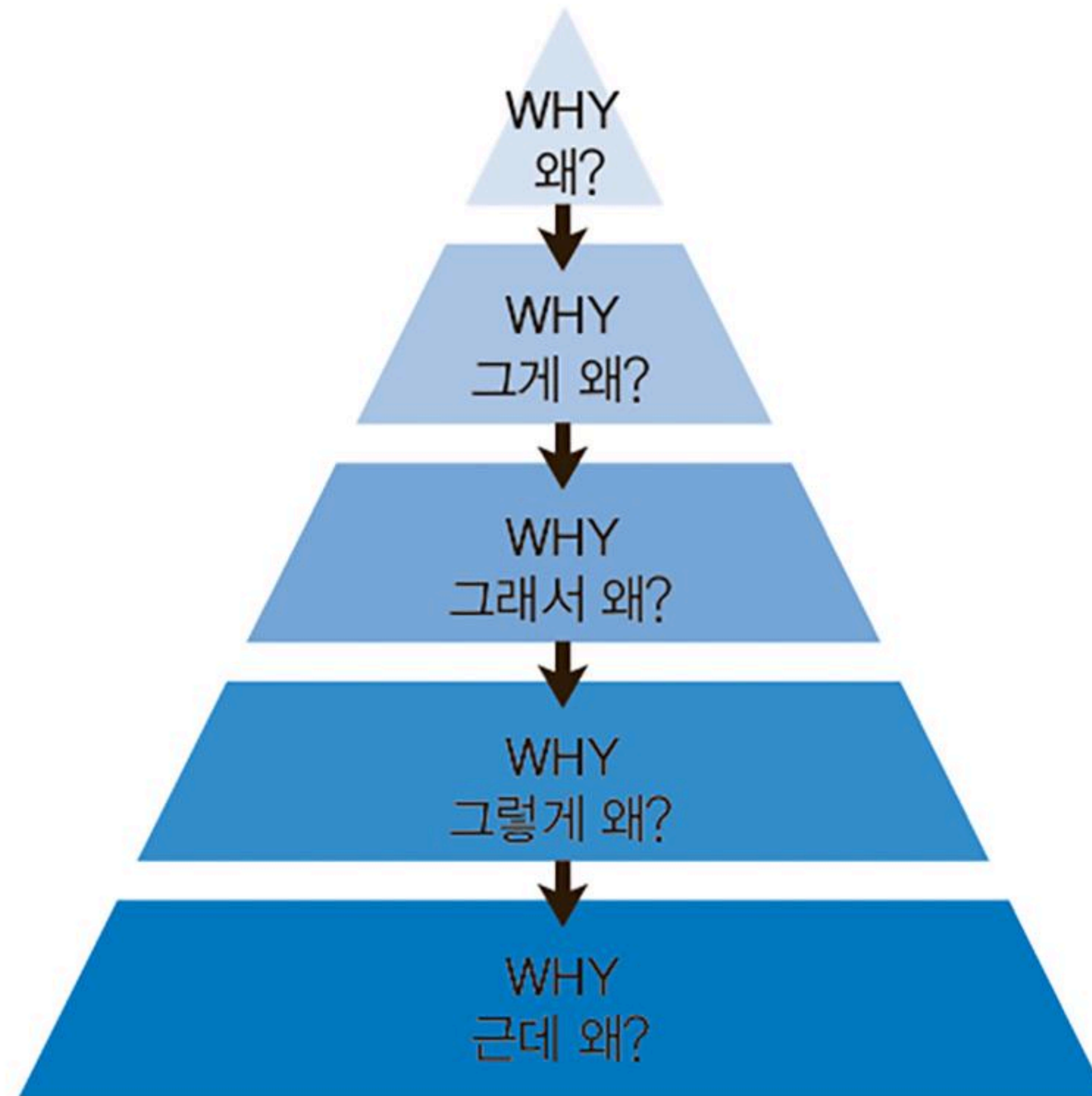
끊임없이 “왜?” 묻는
2살이 되어보자!

5 Whys Method

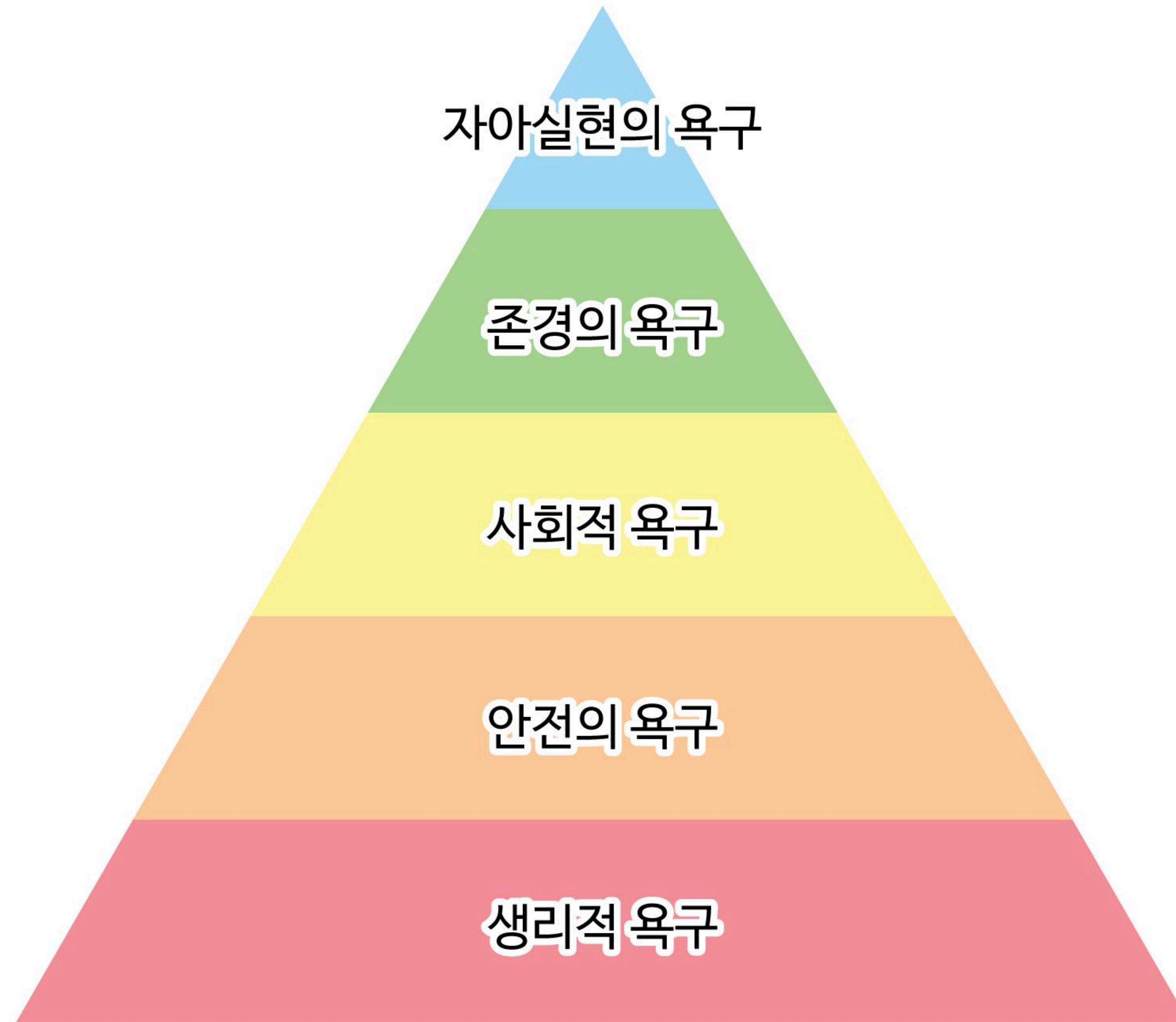
문제의 본질을 파헤칠 때까지
"왜"라고 5번 물어보기!

겉으로 드러난 문제와
진짜 문제는 다르다!

5WHY 개념도



때로는 문제의 본질까지 가보기



[매슬로우의 욕구 위계론]



**제 1원칙 사고법 활용하기!!!
(First Principle Thinking)**

JTBD, Job Stories



JTBD (Jobs-To-Be-Done)

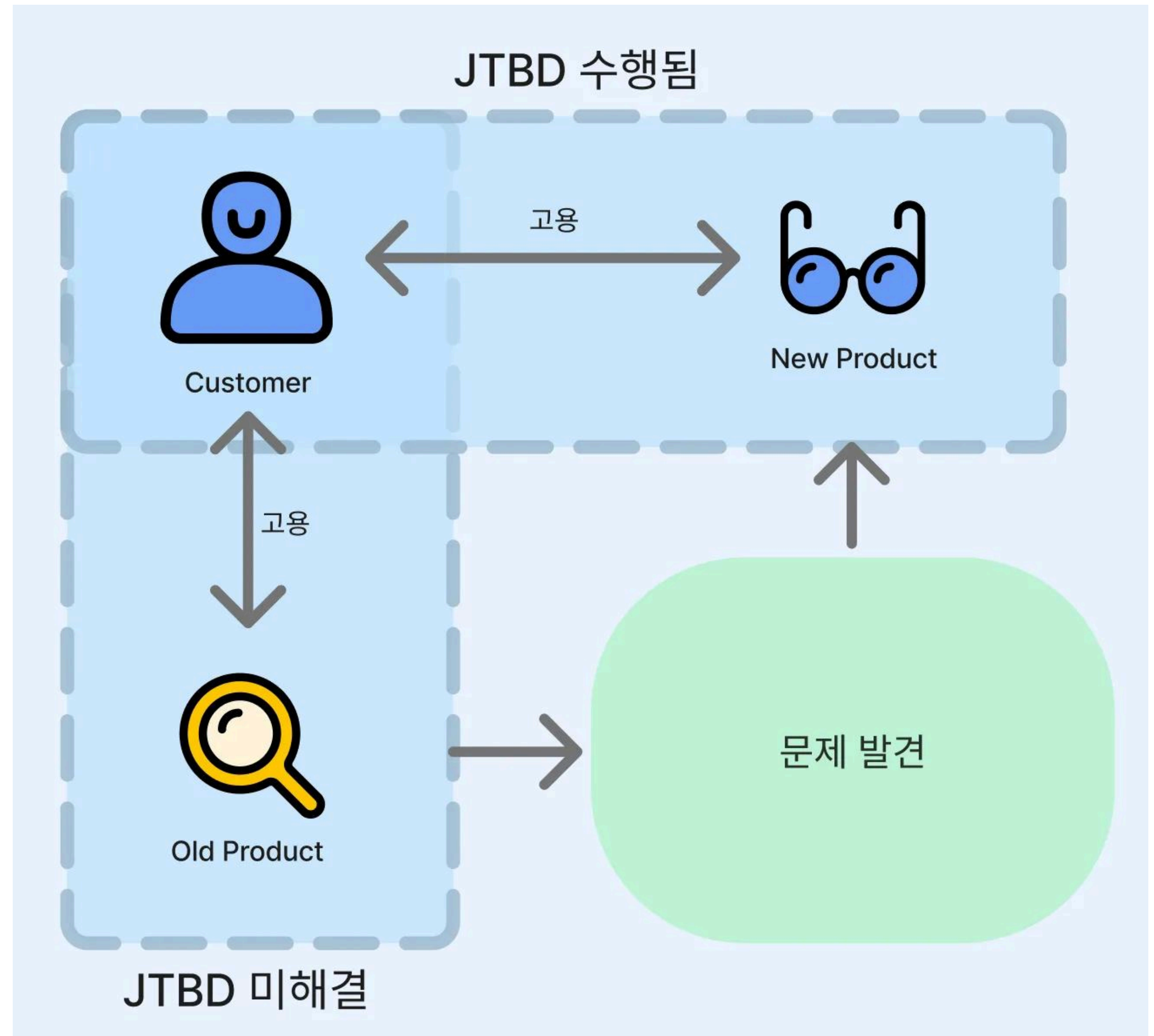
고객은 우리 제품의 기능이
좋아서 사는게 아니다.

그들은 자신의 삶에서 마주친 어떤 문
제나 과업(Job)를 해결하기 위해 우
리 제품을 '고용(Hire)'하는 것이다.

클레이튼 크리스텐스, 하버드 경영대학원 교수

사용자 목표 분석 시스템

사람들은 망치가 좋아서 사는게 아니다.
벽에 못질을 하기 위해 사는 것이다.



<https://techblog.gccompany.co.kr/%EA%B3%A0%EA%B0%9D%EC%9D%84-%EB%B3%B4%EB%8A%94-%EC%83%88%EB%A1%9C%EC%9A%B4-%EB%A0%8C%EC%A6%88-jobs-to-be-done-808c7b72d15b>

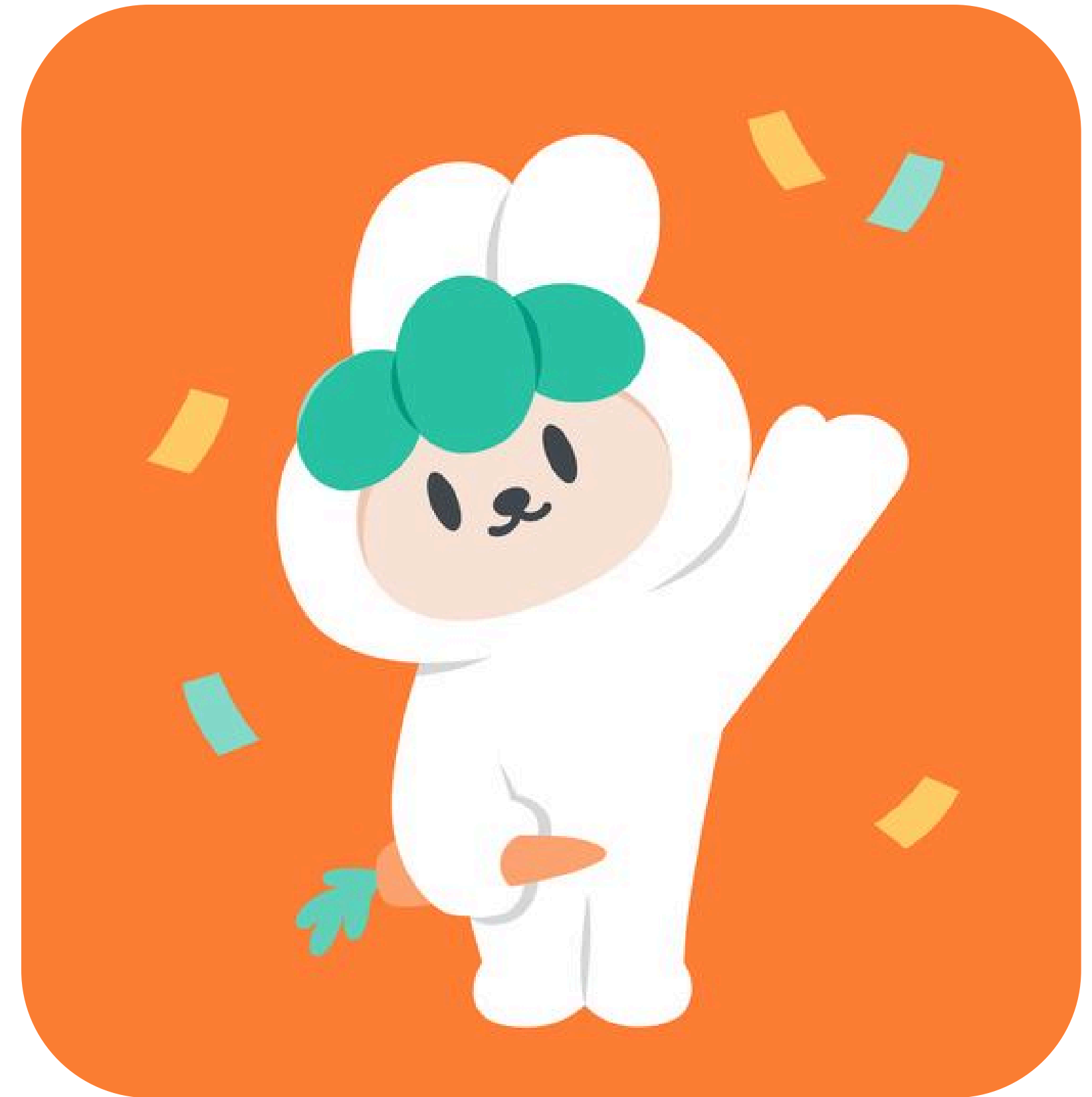
당근마켓 JTBD 예시

사람들은 왜 당근마켓에서 중고거래를 할까?

고객이 해결하려는 진짜 과업/문제(Job):

“모르는 사람과 거래할 때 사기 당하거나, 불쾌한 경험을 하고 싶지 않아.

믿을만한 사람과 빠르고 기분 좋게 거래하고 싶어!”

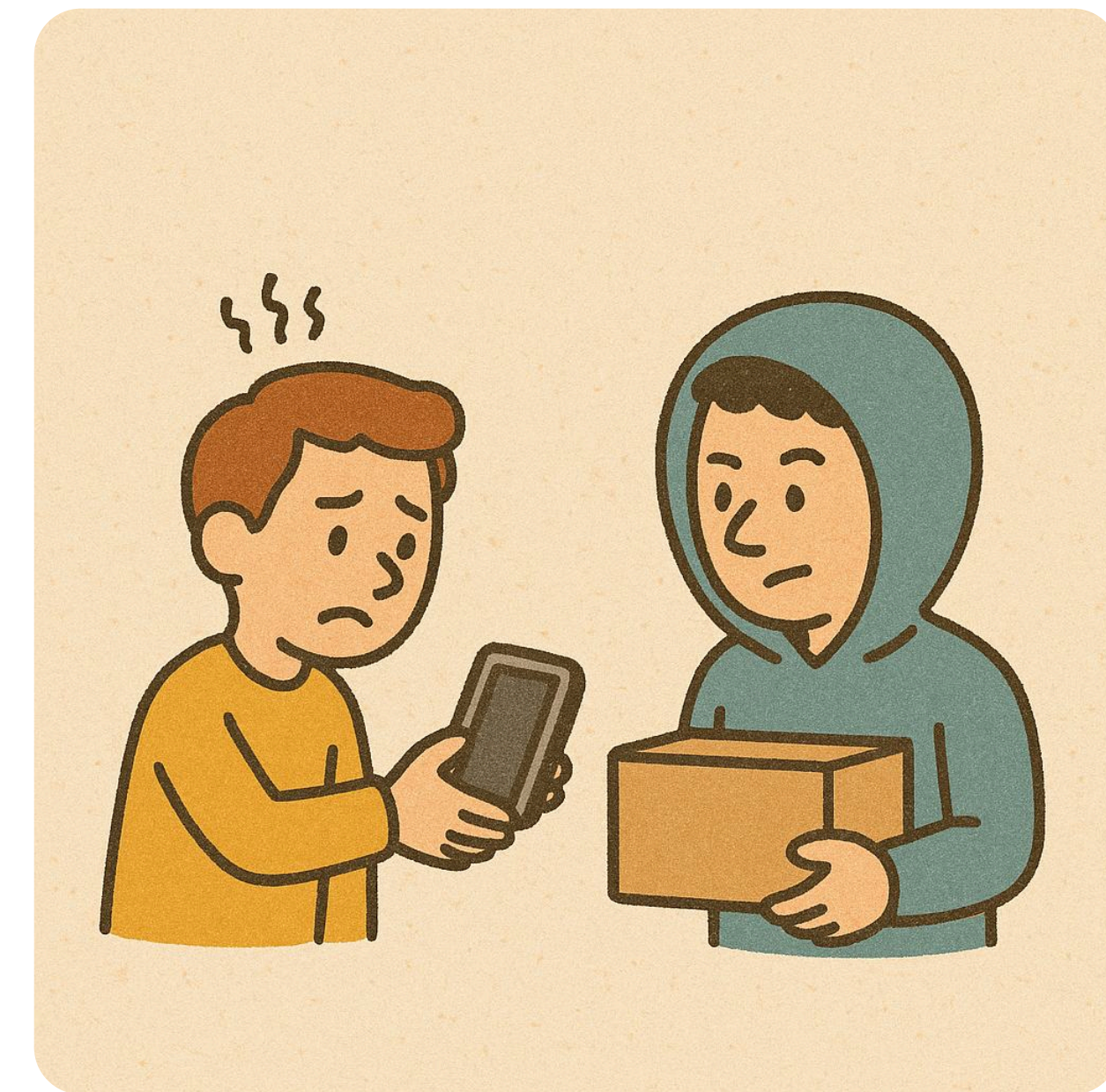


당근의 진짜 경쟁자는?

NOT



BUT

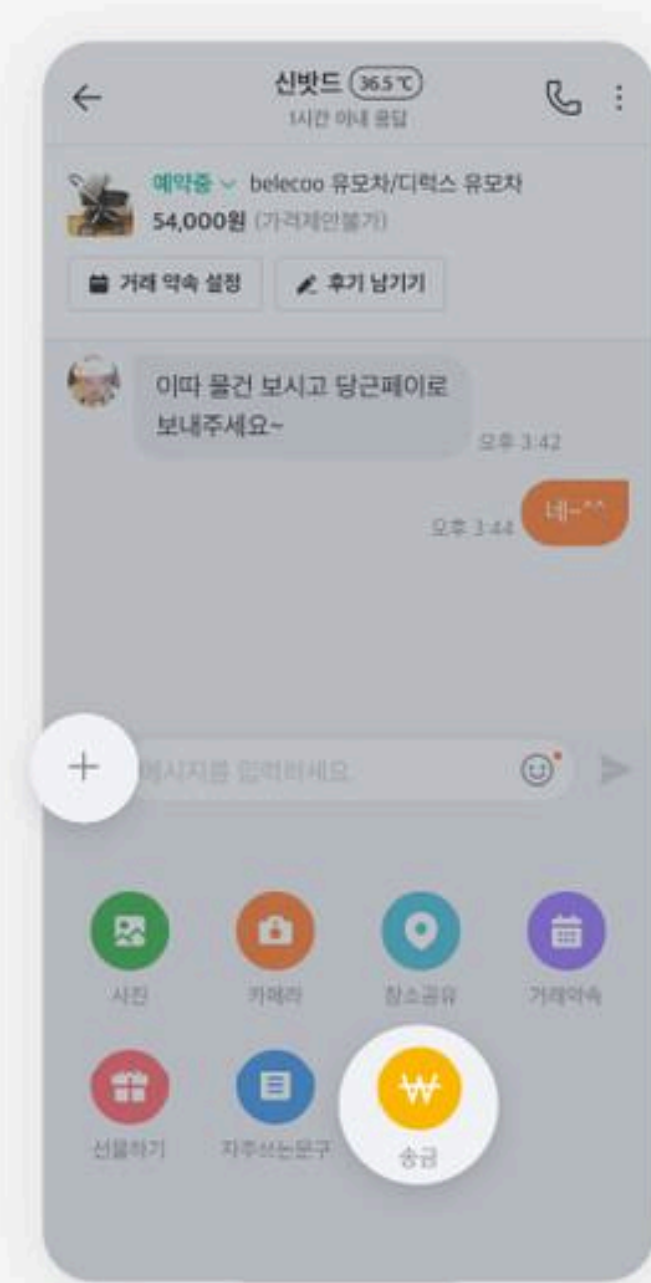


중고 거래에 대한 '불신'과 '불안감' 그 자체
그리고 '그냥 버리기'나 '친구/가족에게 주기'

그래서 나온 기능들 → 매너 온도, 당근페이, 동네 인증



당근페이 제주도 오픈



1 채팅창에서 바로 송금



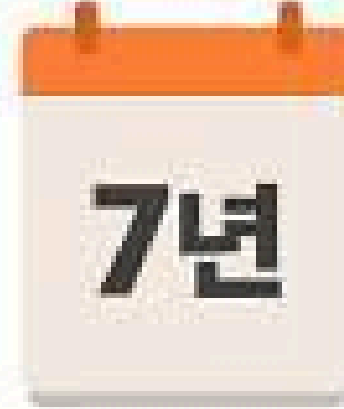
2 간편하고 빠른 이체



3 거래 내역 확인



매너온도 도입된지
2016년 5월 도입



가장 많은 거래 후기
받은 사람

4,900회

가장 많은
나눔을 한 사람

1,856회

전국에 있는
매너온도 99도 이웃



10,098명

전체 가입자 중 0.03%

2022년 한 해 기준

매너온도 99도 이웃이
가장 많이 사는 동네



1위	서울시 강남구	277명
2위	서울시 송파구	198명
3위	성남시 분당구	188명
4위	서울시 강서구	174명
5위	경기도 화성시	172명

Job Story

사용자의 '맥락'과 '동기에 초점을 맞춤

**When ~ 한 상황일때,
I want to ~하고 싶다,
So I can ~한 결과를 얻을 수 있도록**

*뒤에서 나올 User Story와 차별점: User Story는 사용자가 원하는 '기능'에 초점

(회사 내에서) 문제 찾기

회사 내에서 어떤 AI 제품을 만들어야 성공적일까?

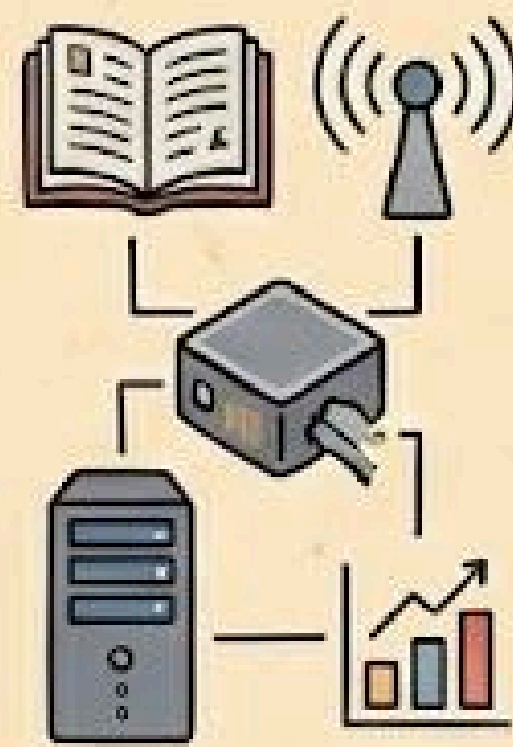
AI 시스템 설계 프레임워크

AI System Design Framework

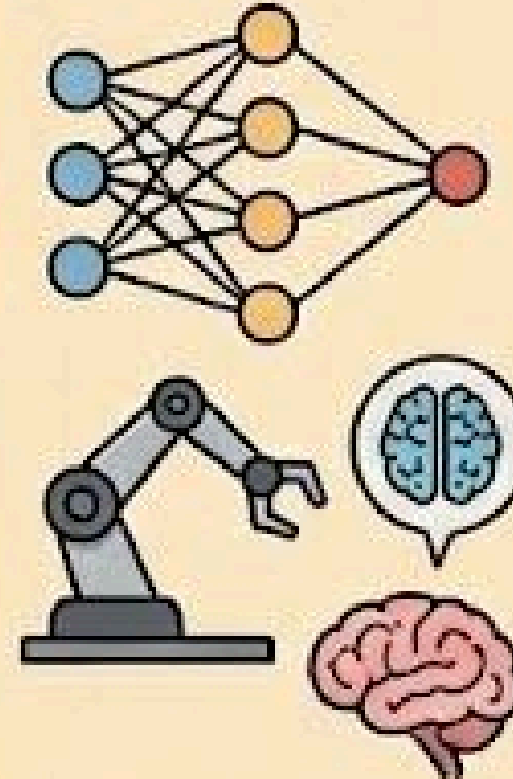
문제 정의
(Problem Definition)



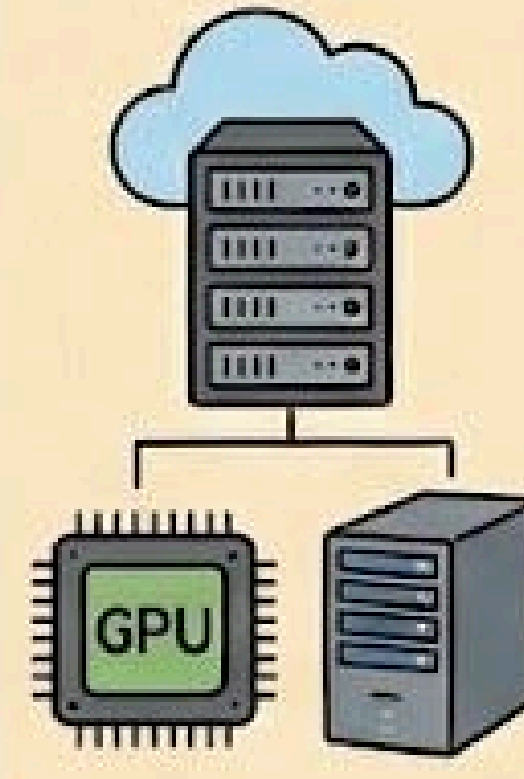
데이터
(Data)



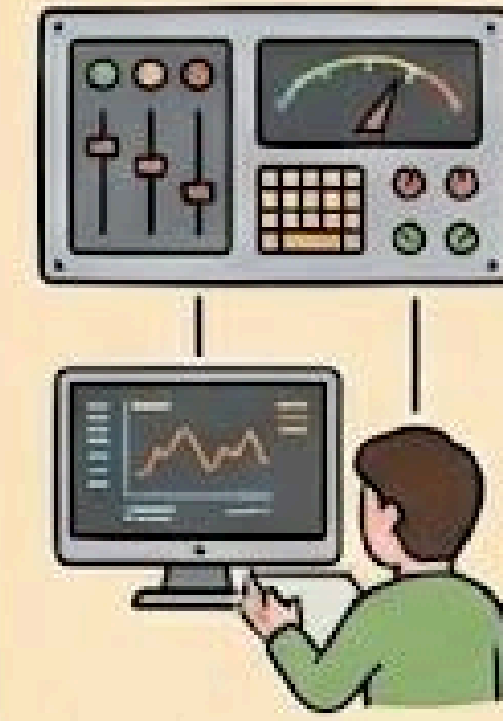
모델
(Model)



인프라
(Infrastructure)



운영
(Operation)



책임
(Responsibility)



1. 문제 정의

- 무엇을 잘하게 만들 것인가
 - 어떤 의사 결정을 바꾸려는가?
 - 사람이 하던 일을 대체하는가, 보조하는가?

→ 문제 정의가 명확하지 않으면, 이후 단계는 모두 기술적 낭비
→ AI로 해결하려는 문제를 LLM과 업무 프로세스 깊하게 논의

2. 데이터 전처리

- 데이터는 재료가 아니라 '역사'
 - 데이터는 조직이 어떻게 일해왔는지의 기록
 - 데이터는 늘 편향, 누락, 관행이 함께 들어있음

- 문제는 데이터가 아니라, 그 데이터를 그대로 쓰는 구조다.
- 데이터 단계는 단순 수집이 아니라, 조직의 과거를 점검하는 단계
- 엑셀이나 PDF 등 LLM에게 먼저 던져주고 정리부터 시키기

3. 모델 선정

- **모델은 주인공이 아니라 '부품'이다**
 - GPT, Claude, Gemini처럼 강력한 API는 누구나 쓸 수 있음
 - 차별점은 모델이 아니라
 - 어떻게 호출하는가
 - 어떤 맥락을 주는가
 - 언제 멈추게 하는가

→ 모델 단계는 기술적 정점이 아니라, 설계의 중간 지점에 불과
→ Planning / Thinking Mode 적절하게 돌아가며 사용하기

4. 인프라

- **AI는 '돌아가는 순간'부터 비용이 됨**
 - 실시간 추론인가, 배치(batch) 처리인가?
 - 클라우드인가, 온프레미스(on-prem)인가?
 - 트래픽이 늘어나면 어떻게 되는가?

→ **인프라는 기술 문제가 아니라 비용과 확장의 문제**

5. 운영

- **AI는 배포되는 순간부터 '관리 대상'이다**
 - 데이터는 변하고, 환경은 달라지고, 모델 성능은 서서히 들림
 - 데이터 드리프트(data drift), 모델 드리프트(model drift)

→ 운영 없는 AI는 브레이크 없는 자동차와 같음

6. 책임 소재

- **설계의 마지막이자, 가장 중요한 단계**
 - AI의 판단을 누가 승인하는가?
 - 오류가 발생하면 누가 중단하는가?
 - 법적, 윤리적 책임은 어디에 있는가?

→ AI 시스템은 인간의 책임 구조 안에 있을 때만 안전함

→ Human-in-the-loop을 어디에 배치할 것인가?

7. 테크니컬 디테일

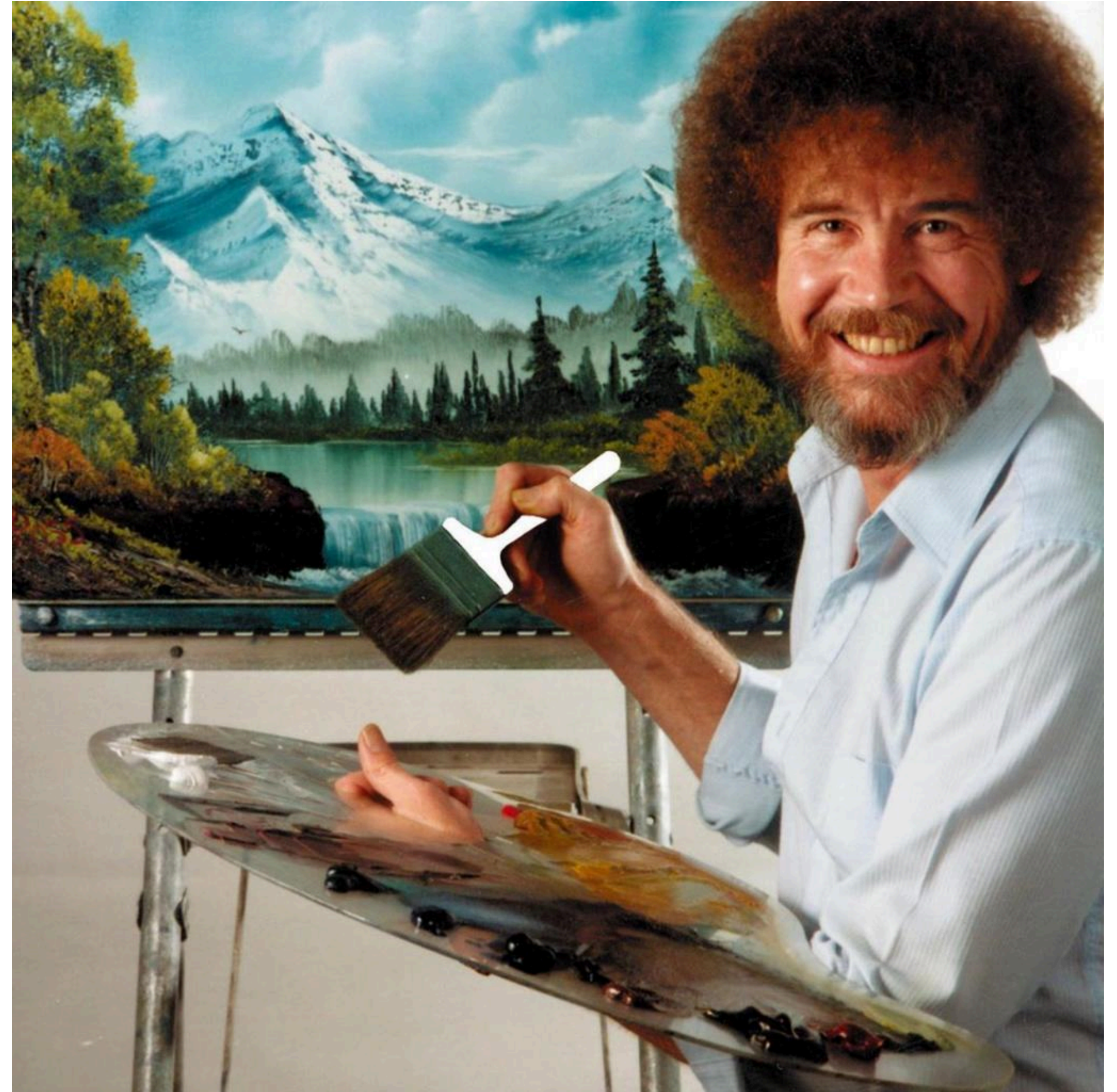
설계 체크리스트

에이전트를 만들 때 이것들을 결정해야 해요:

항목	질문
도구	뭘 할 수 있게 할 건가? (읽기만? 수정도?)
스킬	어떤 지식을 줄 건가?
완료 조건	언제 끝났다고 판단하나?
제한 사항	하면 안 되는 건 뭔가?
출력 형식	결과를 어떤 형태로 받을 건가?
모델	sonnet(빠름/저렴) vs opus(똑똑함)?
최대 턴 수	무한 루프 방지

참 쉽죠? ^^*

모델이 가진 확률적 불확실성을 제품의 제약사항으로 인정하고, 이를 보완할 시스템 (메모리, RAG, 가드레일)을 설계하는 능력

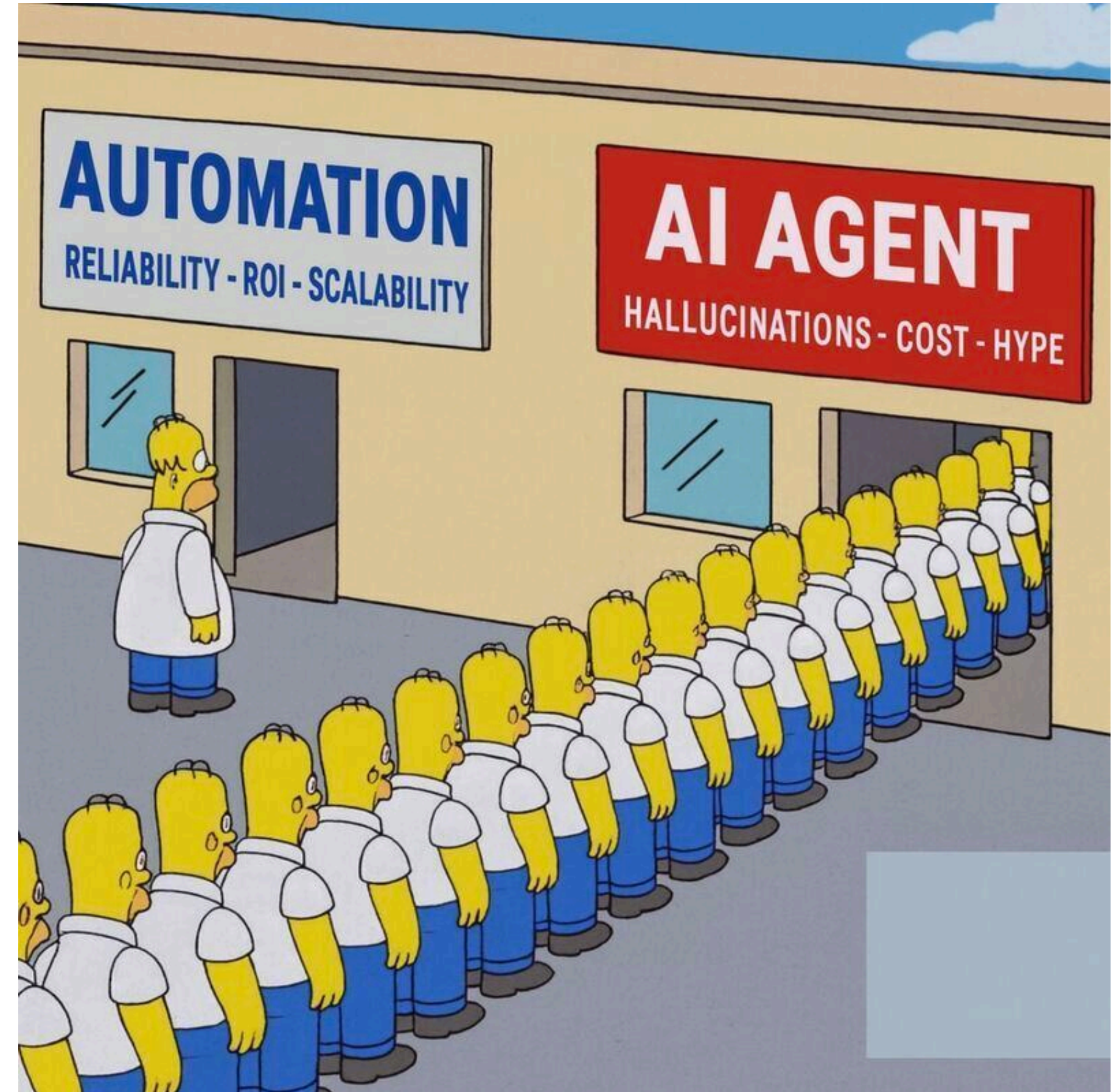


뭘 AI로 만들까?

어떤 제품을 만들지 찾는 법

- 고마찰 (High Friction) 작업
- 고빈도 (High Volume) 작업
- 고맥락 (High Context) 작업

**Tip* - 어디서 맨날 복붙하고 있는가



무엇을 만들까? (1)

- **지식형 검색 허브 (RAG + MCP)**
 - 단순 검색이 아니라 슬랙 대화, 지라 티켓, 노션 문서, 구글 시트 등을 모두 연결한 시스템
 - **기능:** “지난주 마케팅 회의에서 나온 A안의 예산안이 왜 반려됐지?”라고 물으면 관련 문서와 슬랙 로그를 찾아 논리적으로 설명해줌
 - **차별점:** MCP를 활용해 파편화된 사내 툴의 데이터를 실시간으로 읽어옴

무엇을 만들까? (2)

- **데이터 분석 에이전트**
 - 비개발자도 데이터베이스에 직접 질문할 수 있는 도구
 - **기능:** “지난 3개월간 A 상품의 재구매율 추이를 그래프로 그려줘”라고 입력하면 AI가 직접 SQL짜서 데이터 뽑고 시각화까지 완료함
 - **차별점:** 단순 쿼리 생성을 넘어, 데이터의 급격한 변화(Anomaly)를 발견하고 먼저 보고하는 수준까지 구현

무엇을 만들까? (3)

- **운영 자동화 에이전트**
 - 인사, 총무, 법무 등 반복적인 승인 프로세스 처리
 - **기능:** 법인카드 영수증 처리, 연차 신청, 신규 입사자 온보딩 가이드를 에이전트가 주도적으로 수행
 - **차별점:** 사용자가 요청하기 전에 “이번 주 마감인 영수증이 3건 있습니다. 처리할까요?”라고 먼저 말을 걸

핵심 고려사항 & 가드레일

핵심 고려사항 및 가이드라인

1. 데이터 보안 및 권한 관리 (Privacy & Permission)

- “모델은 똑똑하되, 권한은 엄격하게”
- **Role-based Access**
 - 사원이 질문했을 때 팀장급만 볼 수 있는 기밀 문서 내용이 답변에 포함되면 안됨
 - RAG 시스템 설계 시 사용자의 권한을 태깅하여 검색 범위를 제한해야 함
- **Data Masking**
 - 모델로 데이터가 넘어가기 전 PII(개인식별정보)를 자동으로 마스킹하는 레이어가 필요함

핵심 고려사항 및 가이드라인

2. 평가 시스템 구축 (Golden Dataset & Evals)

- “답변이 그럴싸하다”는 주관적 평가 X
- **골든 데이터 셋**
 - 예상 질문과 모범 답변 세트를 최소 100개 이상 확보하여 테스트
- **자동 평가**
 - 새로운 프롬프트나 모델을 적용할 때마다 기존 성능이 유지되는지 LLM-as-a-Judge 방식으로 상시 측정해야 함

핵심 고려사항 및 가드레일

3. 비용 최적화 (Model Tiering)

- 모든 질문에 비싼 모델을 쓸 필요는 없음
- **Tiering**
 - 단순 텍스트 분류나 추출은 SLM(Llama 등)으로, 복잡한 전략 수집은 LLM으로 분기처리하여 토큰 비용 관리하기

도입 전 체크리스트 (비즈니스 파트)

- **수혜자 정의:** 이 프로덕트를 통해 실질적으로 업무가 편해지는 사람이 전체 인원의 몇 %인가?
- **ROI 추정:** 연간 절감되는 인건비나 시간 가치가 시스템 구축 및 토큰 비용을 상회하는가?
- **확장성:** 한 부서에서 성공했을 때 다른 부서나 전사로 즉시 확산 가능한 모델인가?
- **기회비용:** 이 문제를 AI로 해결하지 않았을 때 지불해야 하는 유/무형의 손실은 얼마인가?

AGENT.md

https://docs.google.com/document/d/16m9pwv1HNhMQttxnth4_CL7IaB9Eg-d3-eaf0Ac1jaHM/edit?tab=t.0

나는 [작업명]을 자동화하는 에이전트 시스템을 설계하려고 한다.

작업 개요

- **목적**:
- **입력**:
- **출력**:
- **주요 제약조건**:

내가 원하는 산출물

최종적으로 **하나의 통합 설계서 문서(md 파일)**를 만들고 싶다.
이 문서는 Claude Code에서 구체적인 구현을 할 때 참조할 **계획서** 역할을 한다.

설계서에 포함될 내용

1. **작업 컨텍스트 문서**
 - 배경, 목적, 범위, 입출력 정의, 제약조건, 용어 정의
2. **워크플로우 정의**
 - 단계별 흐름도, 분기조건, 상태 전이
 - LLM 판단 영역과 코드 처리 영역 구분
 - LLM이 직접 수행할 판단/의사결정 단계 명시
 - 각 단계의 성공 기준 및 검증 방법
 - 실패 시 처리 방식 (재시도 / 에스컬레이션 / 스킵)
3. **구현 스펙** (구조 개요만, 상세 내용 x)
 - 폴더 구조
 - CLAUDE.md 핵심 섹션 목록
 - 에이전트 구조 (단일 또는 서브에이전트)
 - 작업 단계별 처리 방식 (에이전트 판단 vs 스크립트)
 - 스킬/스크립트 파일 목록, 역할, 트리거 조건(언제 이 스킬이 호출되는가)
 - 서브에이전트 사용 시: 이름, 역할, 입출력, 데이터 전달 방식
 - 주요 산출물 파일 형식

설계 원칙

[폴더 구조]

에이전트 설정은 `./.claude` 하위에 배치한다.

...



...

[판단과 코드의 역할 분리]

에이전트가 직접 수행	스크립트로 처리
-----	-----
분류, 의사결정, 우선순위 판단	파일 I/O, 데이터 파싱
품질 평가, 정성적 분석	외부 API 호출
컨텍스트 기반 추론	반복 처리, 집계
자연어 생성/요약	정적 분석, 테스트 실행

[검증 패턴]

각 워크플로우 단계에는 반드시 성공 기준을 정의한다. 검증 유형은 산출물 성격에 따라 선택한다.

검증 유형	적용 대상	예시
-----	-----	-----
스키마 검증	구조화된 산출물	필수 필드 존재, 타입 체크
규칙 기반	정량 기준이 있는 작업	항목 수, 글자 수, 필수 섹션 포함
LLM 자기 검증	정성적 판단 작업	요약 품질, 톤 적합성, 누락 여부
사람 검토	고위험 최종 산출물	외부 발송 문서, 의사결정

설계서 작성 시 각 단계마다 다음을 명시한다:

- **성공 기준**: 이 단계의 산출물이 충족해야 할 조건

비밀 하나 알려줌 ^^

·
·
·

AI로 자동화, 에이전트 만드는건 쉬움

회사에서 **사람들이 쓰게 하는게**
100배 더 어려움

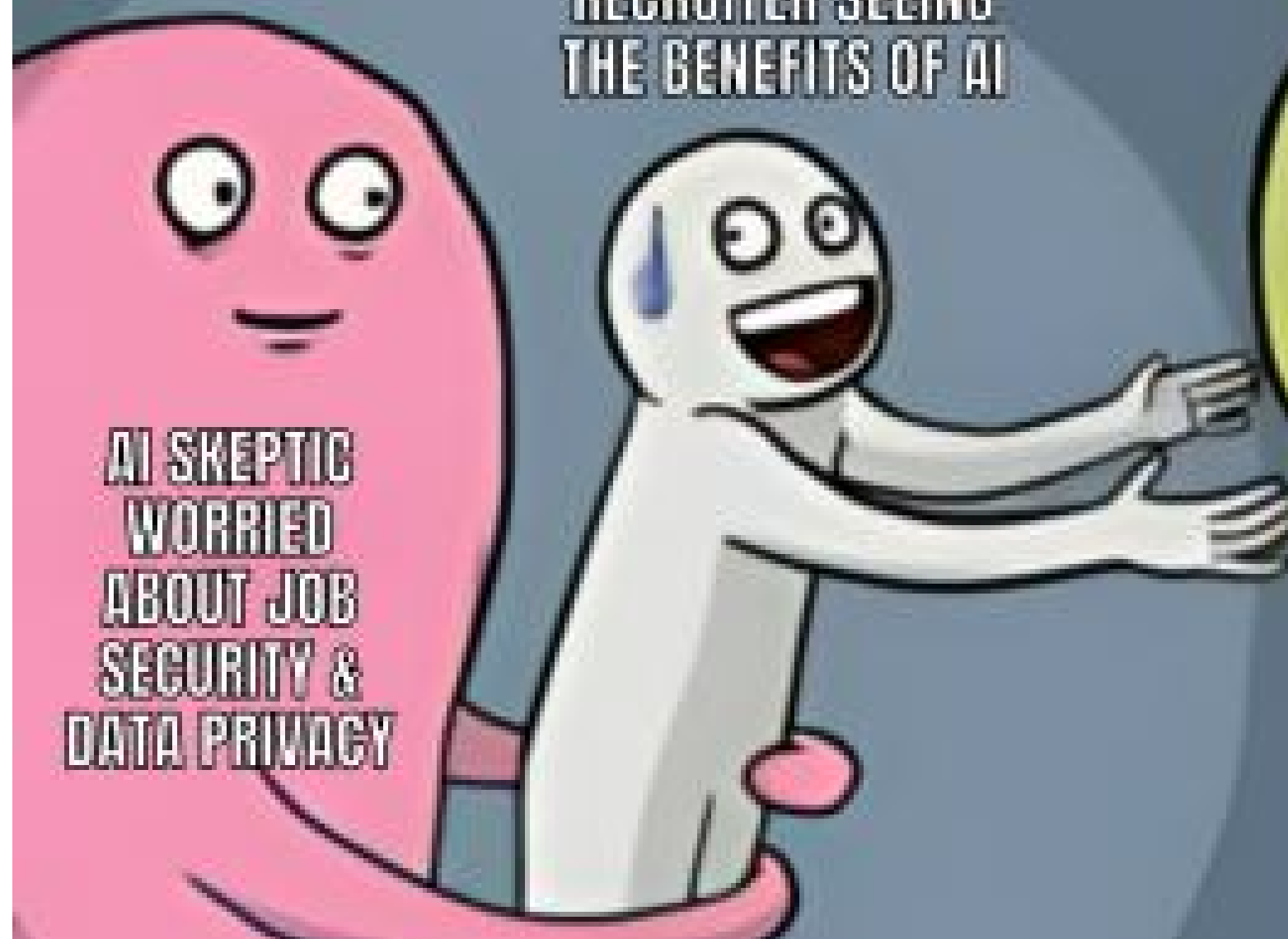


RECRUITER SEEING
THE BENEFITS OF AI



AI WILL MAKE
US ALL MORE
PRODUCTIVE

RECRUITER SEEING
THE BENEFITS OF AI



AI SKEPTIC
WORRIED
ABOUT JOB
SECURITY &
DATA PRIVACY

AI WILL
MAKE US ALL
MORE
PRODUCTIVE

AX

흩어진 데이터와 고비용 인력난, 국내 기업 AX의 '거대한 장벽' 넘으려면

동아일보 | 업데이트 2025.12.24

디지털 전환(DX, Digital Transformation)이 업무의 형식을 바꾼 흐름이었다면, 인공지능 전환(AX, AI Transformation)은 업무의 지능을 고도화하는 작업이다. 디지털 전환은 약 10년에 걸쳐 클라우드 전환과 전사적 자원관리(ERP)의 도입, 데이터베이스 구축 측면에서 진행됐다. 반면 인공지능 전환은 1년에서 3년 이내에 이뤄지고 있으며, 자체 데이터를 기반으로 AI를 학습하고, 대규모 언어 모델 도입을 통한 업무 방식의 고도화하는 측면으로 접근한다. 전자는 프로세스의 관리나 효율 측면이었는데, 후자는 의사결정의 지능화와 전략화에 가깝다.

AX 도입 하다가 속이 터집니다 속이... ^^*





AI 팀은 결과를 냈지만, **현장은 쓰지 않음**
현장은 필요하다고 하지만, 요구사항 모호함
경영진은 관심있지만, 책임 구조는 불명확함



처음부터 전사적 시스템을 바꾸기 → **100% FAIL**
특정 팀의 가장 귀찮은 업무 하나 자동화 → **SMALL WIN**

결국 AX 도입 문화 조성을 위한
설득과 커뮤니케이션이 가장 중요

페르소나 정의

우리는 정확히 '누구'의 문제를 해결하나?

일반 B2C 대상 프로덕트 페르소나

**모두를 위한 서비스는 결국
아무도 쓰지 않는 서비스가 된다.**

**“가장 흔한 실수는 타겟을 너무 넓게 잡는 것이다.
성공적인 타겟팅은 주변에서 ‘편협하다’고 오해할 정도로
거의 우스꽝스러울만큼 집요하고 명확해야 한다.”**

“When it comes to the question of the target customer, the most common mistake is the definition is too broad. It must be almost *comically narrow*, to the point you may be misunderstood for such a narrow focus.”

Andy Johns
former growth lead at Facebook, Twitter, and Quora

목표 고객을 좁힐수록
제품은 날카로워 지고,
메세지를 분명해지며,
고객 반응은 폭발한다.

페르소나(Persona)

- 서비스를 사용하는 주요 고객을 대표하는 인물 = 메인 타겟
- 실제 사용자 정보를 바탕으로 어떤 인물을 구체적으로 정의
- 우리 서비스의 초기 광팬이 될 Early Adopter 페르소나
- Ideal Customer Profile (ICP)라고도 함

- 실제로 돈을 지불한/할 고객의 정보로 분석하기!
- 고객 페르소나는 여러명 일 수 있음 (코호트)
- 초기 ICP는 틀릴 확률 높음 → 점차 진화

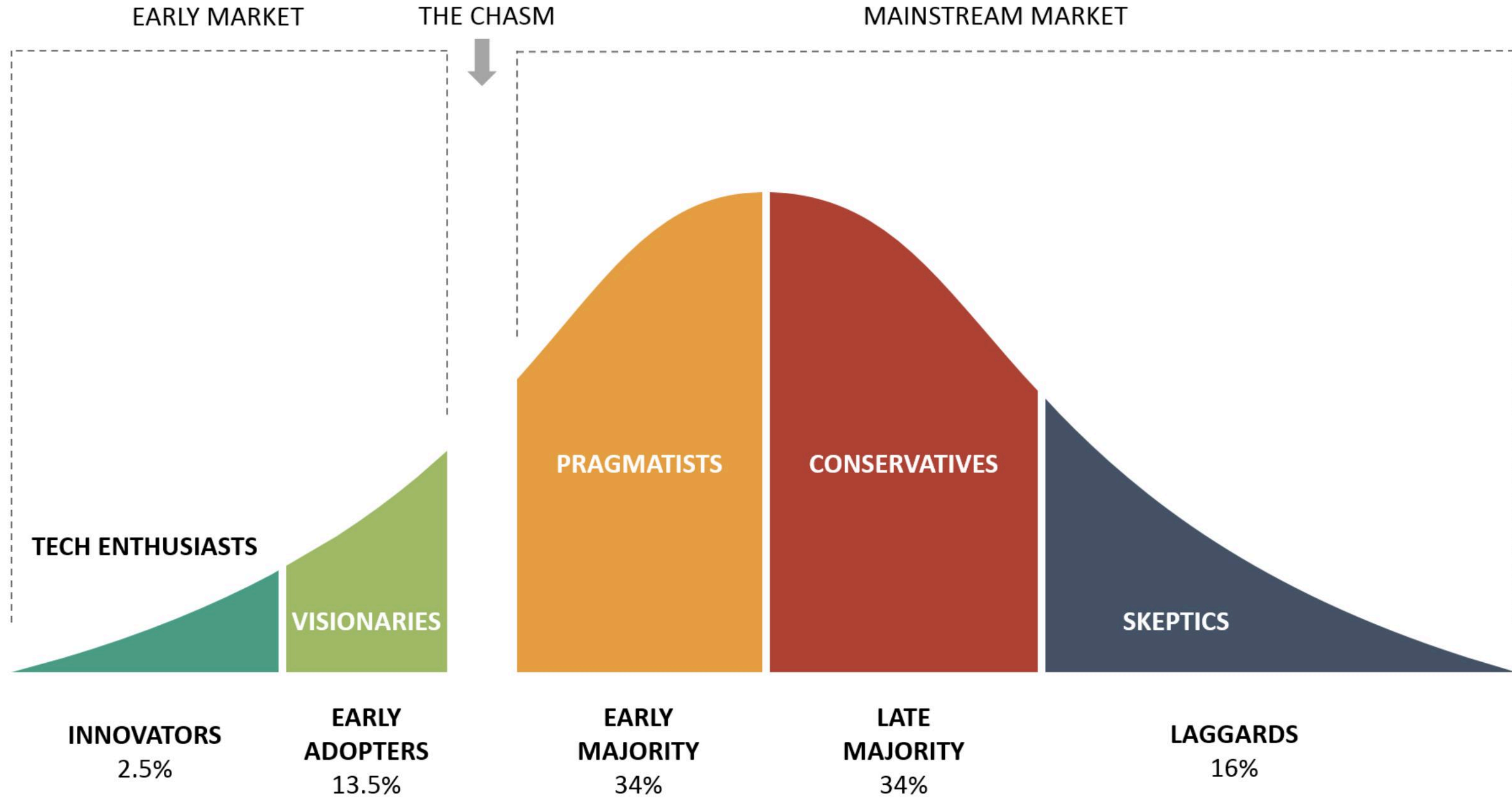
Examples of super-specific who

Company	Super-specific who
	25-to-35-year-old women in cities
	Gamers on PCs playing Final Fantasy XIV
	Attractive, party-loving college students in L.A.
	Online DVD fanatics
	Young lower-tier influencers on Facebook or Instagram

	Designers interested in photography with a large Twitter following
	Conference-goers traveling to a city where hotels are booked up
	Independent restaurant owners in suburban markets
	Successful veteran online newsletter writers
	Young tech employees in San Francisco
	Successful professionals in Silicon Valley
	Millennials who don't have a lot to invest

첫 고객 선정(Beachhead)

- 가장 열정적으로 반응할 소수의 첫 고객
- 이후 점진적으로 고객군 확장



Persona



"ENTP"

김세은

- 성별 - 여자
- 나이 - 28세
- 직업 - 3년차 콘텐츠 마케터
- 연봉 - 3300~3500
- 거주지역 - 신림
- 주거형태 - 오피스텔 원룸
- 주거비용 - 3000 / 60
- 가족구성원 - 자취
- 여유 소비 금액 - 60만원

취미

- 맛집 탐방
- 분위기 좋은 카페가기
- 인스타그램 보기

정보이용방법

- 인스타그램
- 유튜브

생활 패턴

- 많이 먹기 보다는 하나를 먹어도 맛있는 걸 먹는다.
- 디저트 류를 좋아한다.
- 배가 불러도, 맛있는 디저트 하나는 꼭 먹는 편이다.
- 핫플레이스를 좋아하지만, 다른사람과 차별화되기를 원한다.

성격

- 색다른 아이디어가 풍부하며 새로운 가치를 발견하는 일에 만족감을 느낀다.
- 남들보다 먼저하는것을 좋아하며, 그것을 공유해주는것을 즐김.

< Situation

28세의 3년차 콘텐츠 마케터.
강남 출퇴근을 위해 신림에 오피스텔을 구해서 자취 중.

유행에 민감해서 SNS핫플레이스는 모두 찾아다니는 편.
특히나 빵을 좋아해서,
유명 빵집의 대표메뉴는 거의 다 먹어봄.

하지만, 다른 사람과 차별화를 원해서 신상 맛집과 새로운 빵을 찾아다닌다.

오늘도 SNS를 탐색하며 '어디가 새롭게 뜨는 빵 맛집일까' 검색중..

크리스마스에 특별한 케이크를 주문해서 예약하고 싶은데,
'어떤 브랜드의 케이크가 이쁘면서 맛있을까'가 최근 관심사!

Personality

유행에 둔감

유행에 민감

바쁜

여유로운

수동적

주도적

비활동적

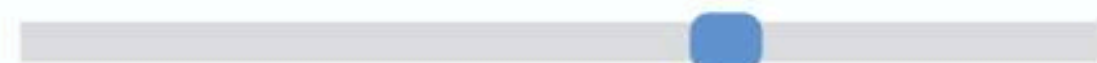
활동적

[Bio]



[Behaviors/Context(행동/맥락)]

[불편을 느끼는 지점(Pain Points)]



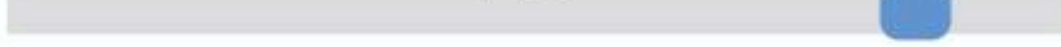
low

high

[목표/니즈(Goals/Needs)]

[필요한 기능(Featured Needed)]

페르소나 템플릿 예시

 [Bio] 이름 : 김나래 나이 : 24 성별 : 여 직업 : 대학생 사용 서비스 : 지그재그, 에이블리, 브랜디, 스타일 쉐어 등	[Behaviors/Context(행동/맥락)] - 라이브 방송 알림이 뜨면 습관처럼 들어가보게 된다. - 한번 스타일도 맞고 신뢰가가는 셀러를 찾으면 추천해주는 옷들 위주로 사게된다 - 단골로 보는 셀러가 3~5개 정도 있다	[불편을 느끼는 지점(Pain Points)] - 온라인에서 사이즈도 틀리고 컬러도 틀린적이 많아 매번 알면서도 속는다. - 인스타그램에서 DM이나 댓글로 신청하고 사는 것이 오래걸리고 번거로울 때가 있다.
가격 민감도  SNS 활용도  온라인 쇼핑 빈도  low high	[목표/니즈(Goals/Needs)] - 좋은 옷을 적당한 가격에 속지않고 사고 싶다. - 옷을 하나 사면 여러 옷들과 매칭해서 가성비가 높아질 수 있는 옷을 사고 싶다 - 옷을 매칭하기가 어려운데 누가 좀 알려주면 좋겠다. - 간편하게 결제와 배송이 이루어지면 좋겠다.	[필요한 기능(Featured Needed)] - 영상을 통한 판매를 통해 정확하고 신뢰있는 정보를 전달 - 자신에게 맞는 옷을 추천 - 간편 결제 - 셀러를 중심으로한 커뮤니티

올리브영 타겟 페르소나



기본 정보	이름: 김지연 나이: 28세 직업: 대기업 마케팅팀 사원
라이프 스타일	• 도시에 거주하며 최신 뷰티 트렌드에 민감함 • 바쁜 일과 후나 주말에는 자신만의 뷰티 루틴을 실천함 • 건강하고 빛나는 피부를 위해 스킨케어 제품에 적극적으로 투자
주요 관심사	• 트렌드: 신상 화장품 및 SNS에서 유행하는 제품에 관심이 많음 • 가성비: 좋은 품질과 합리적인 가격의 균형을 중요하게 생각함 • 웰니스: 뷰티뿐만 아니라 건강(이너뷰티 등)까지 함께 챙길 수 있는 제품을 선호
구매 행동	• 화해, 글로우픽 등 뷰티 앱의 리뷰와 SNS 추천을 신뢰하고 자주 구매 • 오프라인 매장에서 직접 테스트해보거나, 온라인몰의 빠른 배송(오늘드림) 이용 • 다양한 샘플을 사용해 본 후 자신의 피부에 맞는 제품을 최종 결정함
브랜드 페르소나	• "친근하고 트렌디한 언니/친구" • 최신 유행에 밝고 친근하게 소통하며, 고객의 즐거운 쇼핑 경험과 정서적 유대감 중요시 하는 페르소나

마켓컬리 타겟 페르소나



기본 정보	이름: 이수현 나이: 35세 직업: 워킹맘 (IT회사 마케팅 매니저)
라이프 스타일	• 평일에는 워킹맘, 주말엔 가족과 함께 건강하고 신선한 음식을 즐기려고 함 • 품질 좋은 식재료와 간단한 레시피 선호함 • 바쁜 일상과 야근 후에도 온라인으로 편리하게 장을 봄
주요 관심사	• 신선도: 가족의 건강을 위해 품질 좋은 유기농/신선 식재료를 최우선으로 고려함. • 편리함: 바쁜 일상 속 장보기 시간을 줄여주는 빠른배송을 선호 • 트렌드: 비건, 오가닉 등 건강과 관련된 최신 식품 트렌드에 관심
구매 행동	• 신선도와 품질이 포장된 유기농, 친환경 제품을 우선적으로 선택 • 당일배송 등 빠른 배송이 가능한지 확인 후 구매함 • 제공되는 레시피를 참고하여 관련 상품 함께 구매하는 경향 • 다른 사용자들의 리뷰를 참고하여 최종 구매를 결정함
브랜드 페르소나	• “깐깐하고 신뢰할 수 있는 큐레이터” • 바쁜 3040 워킹맘을 위해 전문가가 엄선한 고품질의 상품과 편리한 라이프스타일을 제안하는 페르소나

당신의 아이디어에 대해 이야기하지 말고
고객의 삶과 문제에 대해 이야기하라.



고객이 하는 말이 아닌 하는 행동에 집중하라

“Pay attention to what users
do, not what they say.”

Jakob Nielsen, a legendary UX Researcher

How Might We...? (HMW)

우리 제품/서비스가 어떻게 하면 고객의 과업/불편함을 해소해 줄 수 있을까...?

How Might We + **Intended Action**
(as an action verb) + **For** + **Potential User**
(as the subject) + **So That** + **Desired Outcome**

For example:

- How might we **provide healthier meals** for **teenagers** so that **they stay away from junk food**?



사내 직원 대상 AI 프로덕트 페르소나

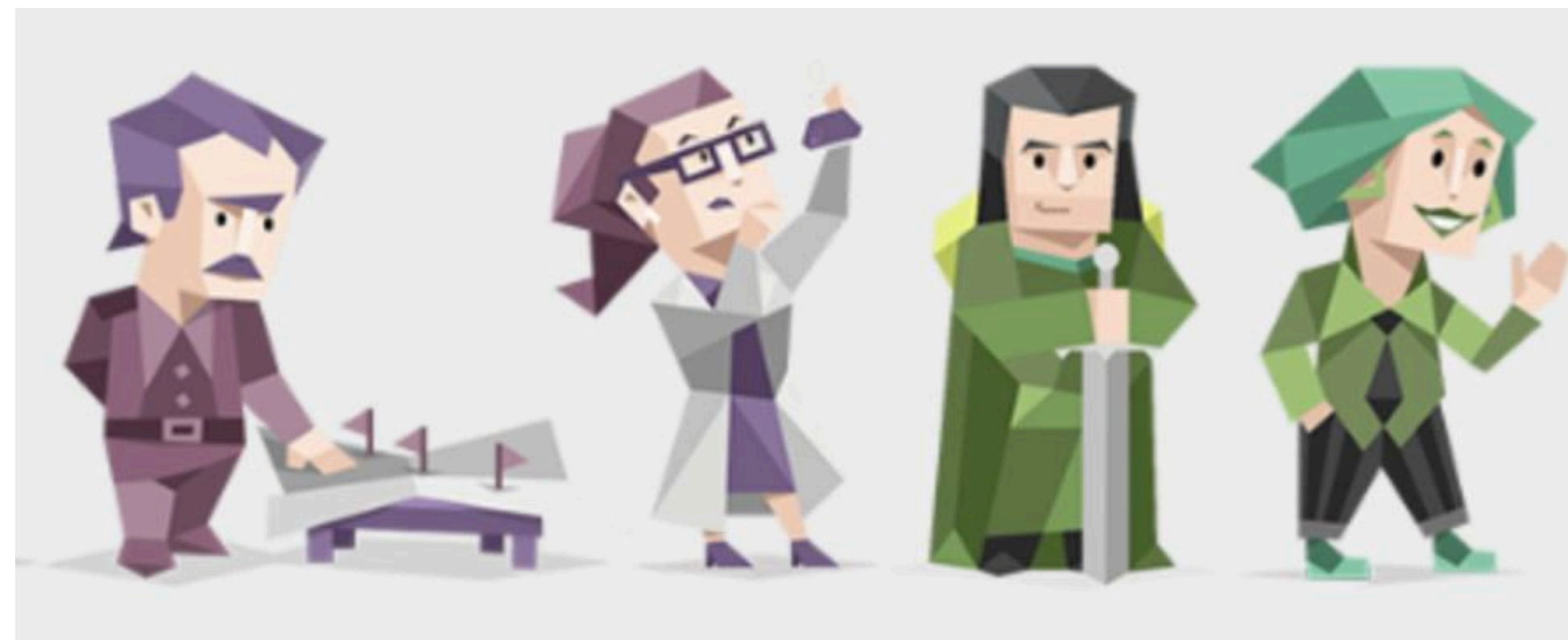
사내 직원 대상 AI 제품 만들때

구분	대중 대상 (External)	사내 직원 대상 (Internal)
핵심 목표	매력도, 전환, 리텐션 (재미/편의)	효율성, 정확성, 워크플로우 통합
사용 동기	자발적 사용 (취향과 필요)	필연적 사용 (업무 완수가 목적)
페르소나 분류 기준	인구통계학, 라이프스타일, 취향	직무(Job Function), 권한(Permission), 툴 스택
AI에 대한 기대	"나를 놀라게 해줘" (놀라움/창의성)	"내 일을 확실히 끝내줘" (신뢰/안정성)
데이터 범위	개인 데이터 및 일반적 상식	사내 기밀 데이터 및 복잡한 업무 맥락

새로운 페르소나 고려사항

1. AI 숙련도 (AI Literacy)
2. 도구 활용 능력 (Tool Proficiency)
3. 데이터 접근 권한 (Data Accessibility)

사내 AI 제품을 만들 때는 '사람' 그 자체보다
그 사람이 처한 '**업무 환경**'을 **페르소나화** 하는게
더 정확할 수 있음 (Workflow)



직무 기반 세그멘테이션

- 1. The Researcher** (예: 법무, 기획)
- 2. The Calculator** (예: 재무, 데이터)
- 3. The Communicator** (예: 영업, CS)

페르소나 예시

- 페르소나: 정보 과부하에 빠진 신규 입사자 김신입

항목	상세 내용
주요 직무	마케팅팀 신입 사원 (온보딩 중)
업무 환경	사내 위키, 노션, 슬랙 채널이 너무 많아 어디에 정보가 있는지 모름.
AI 숙련도	ChatGPT를 써보긴 했으나, 회사 데이터에도 물어봐도 되는지 불안해함.
핵심 니즈	"팀장님께 물어보기 민망한 기초적인 규정(예: 연차 신청법, 폰트 가이드)을 빨리 찾고 싶음."
AI 가치 제안	흩어진 문서에서 답을 찾아주고, 해당 규정으로 바로 가는 링크까지 제공.
주의 사항	신입 사원이므로 팀의 '비공개 전략 문서'는 답변에서 제외되어야 함 (보안).

AI 회의론자

가장 주의해야 할 페르소나는 **AI 회의론자**
→ 한 번 AI로 업무가 해결 안 되면 다시 잘 안씀



살아남는 AI 프로덕트 만들기

90%의 AI 프로덕트가 망하는 시장에서 살아남는 법

거대 LLM이 업데이트 한 번 하면
모조리 망하는 AI 스타트업들....



Wrapper vs. System

AI Wrapper

- LLM API에 살짝 UI만 씌운 제품
- 예) 단순 요약기, 단순 챗봇

Deep Integration

- 독자적인 RAG 아키텍처, 복잡한 에이전틱 워크플로우, 하이브리드 모델(LLM+전통적 알고리즘)을 갖춘 제품

“오늘 나온 AI 모델이 가장 멍청하다”

“This is the worst the models will ever be.”

-OpenAI 부사장 *Kevin Weil*

현재의 고점(Local Maxima)에 갇히지 말 것

→ 장기적으로는 모델 자체가 검색과 추론을
더 완벽하게 수행하게 될 것

**오늘의 모델 성능이 아닌, 6~12개월 뒤
모델이 도달할 지점을 예상하고 제품 설계할 것**



LLM Wrapper



AI Agent

IDE vs. CLI

IDE (Integrated Development Environment)

- VS Code, Cursor처럼 코드 편집, 디버깅, 실행을 한 곳에서 하는 복잡한 소프트웨어

CLI (Command Line Interface)

- 터미널이나 프롬프트처럼 명령어를 직접 타이핑하는 환경

JS todo.js

Cursor Settings

Untitled-1

Search

can you name this file

1 | %L to chat, %K to generate

2

3 ```Untitled-1

4 // Start Generation Here

5

6

7 fn is_prime(n: u32) -> bool {

8 if n <= 1 {

9 return false;

10 }

11 for i in 2..(n as f64).sqrt() as u32 + 1 {

12 if n % i == 0 {

13 return false;

14 }

15 }

16 true

17 }

18

19 fn main() {

20 let mut input = String::new();

21 println!("Enter a number: ");

22 std::io::stdin().read_line(&mut input).expect("Failed to read line");

23 let num: u32 = match input.trim().parse() {

24 Ok(num) => num,

25 Err(_) => panic!("Please type a number!"),

26 };

27 if is_prime(num) {

28 println!("{}", num);

29 } else {

30 println!("{}", num);

31 }

32 }

33

34

35

36

37

CHAT

Based on the code, I would suggest naming this file prime_checker.rs . This name accurately reflects the purpose of the code, which is to check whether a given number is prime or not.

Ask followup (%Y), ↑ to select

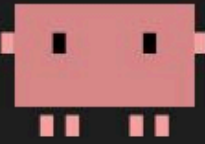
^ codestral @ Mention ↩ chat ⌘ ↩ codebase ^

Competitive Analysis — claude TMPDIR=/var/folders/5f/70kn69md0ksftml0...

Last login: Mon Oct 20 12:25:46 on ttys021

[(llm-evals-3.13) ttorres@Terasas-Mini ~ % cd Documents/Competitive\ Analysis]

[(llm-evals-3.13) ttorres@Terasas-Mini Competitive Analysis % claude]



Claude Code v2.0.24

Sonnet 4.5 · Claude Max

/Users/ttorres/Documents/Competitive Analysis

> hi claude

• Hi Teresa! How can I help you today?

>

? for shortcuts Thinking off (tab to toggle)

Why CLI?

- **집중력 유지**
 - IDE는 수많은 파일과 버튼 때문에 “지금 어디 있지?”를 고민하게 만듦
 - CLI는 오직 “현재 진행 상황”에만 집중하게 함
- **로컬 환경의 힘**
 - 에이전트가 내 컴퓨터의 DB에 직접 들어가 문제를 해결하고 테스트할 수 있는 ‘자유’가 더 많음

SaaS의 종말?

지금은 모든 회사가 똑같은 SaaS(예: 슬랙, 노션)를 쓰지만,
**곧 각 회사에 맞게 코드가 실시간으로 수정되는
전용 소프트웨어를 쓰게 될 것**

⇒ 에이전트가 본사의 업데이트 사항을
내 환경에 맞춰 자동으로 합쳐줌

앱의 종말?

- 식단 관리 앱: AI에게 음식 사진 주고 운동 짜달라고 하면 됨
- To Do 앱: "내일 아침에 알려줘"라고 명령해놓으면 됨

→ 살아남는 앱: 실제 센서나 하드웨어 제어권이 필요한 특수 앱

데이터를 관리하고 보여주는 수준의 앱의 80%는 사라질 것



코드 직접 타이핑 X
에이전트 지휘/검증 O

**사장이 실무를 하는 회사는 AI 시대 훨씬 강력해지고
사장이 실무를 하지 않는 회사는 없어질 것**

가치 사슬 포지셔닝 (Value Chain Positioning)

Horizontal (수평적)

- 범용적인 기능 제공 (오픈 AI, Google)
- 정면 대결했다가 망하기 딱 좋음

Vertical (수직적)

- 특정 산업 (법률, 의료, 제조)의 문제 해결
- 10배 나은 경험을 제공하는 전문 분야

살아남는 AI 프로젝트 만들기

1. 독점적 데이터 해자
2. 워크 플로우 고착화
3. 에이전틱 복잡성
4. 신뢰와 가드레일

AI 경쟁사 분석 체크리스트

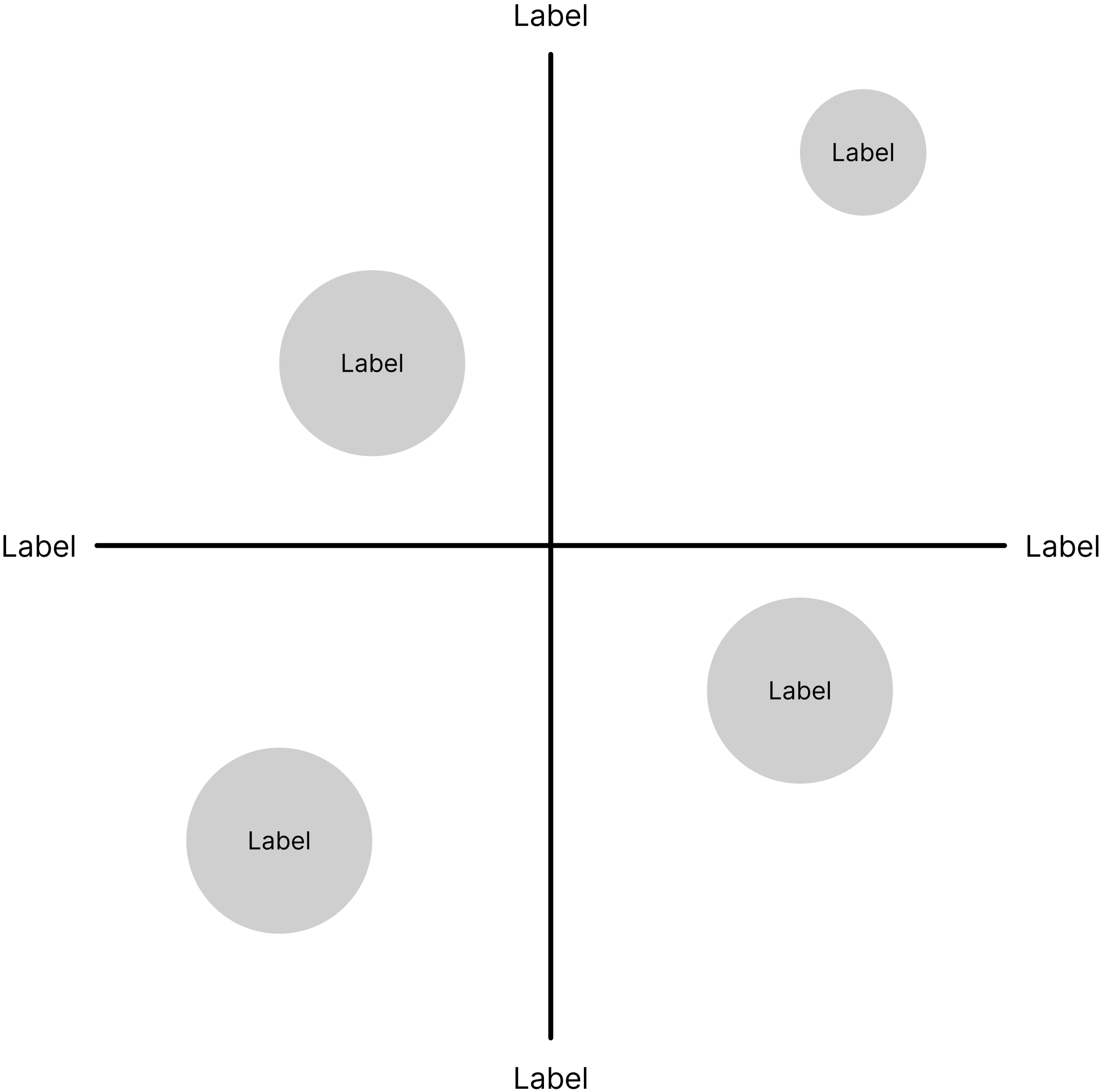
1. 지능의 한계
2. 데이터의 원천
3. 전환 비용
4. 연결성 (MCP/CLI)

경쟁사 분석

	A사	B사	C사
타겟 유저			
핵심 가치			
장점/경쟁우위			
단점/불편한 점			

경쟁사 포지셔닝 맵

고객이 진짜로 중요하게 생각하는
“핵심 가치”를 축으로 삼아 포지셔닝





국내 이커머스 시장 포지셔닝 맵



AS IS - TO BE

그래서 무엇을 만들 것인가?

**As-Is(현재)와 To-Be(이상) 사이 갭이
바로 우리 제품의 '핵심 가치'이다.**

타협 없는 사용자 경험 AS IS - TO BE

AS IS

현재의 상태를 이해하고 문제점을 식별하는 단계

문제점 파악 및 사용자 니즈 파악

필요한 서비스 제공 / 개선 필요성 체크



TO BE

개선된 상태를 설정하고 목표를 정하는 단계

더 나은 경험을 얻을 수 있도록 설계하고 구체화

사용자들에게 가치를 제공할 수 있는지를 결정
/ 해결책 제공

CAASTUDIO

As-Is (택시 잡기의 고통)

1. 기다림: 길가에 나가 하염없이 손을 흔든다
2. 불확실성: 택시가 올지, 언제 올지, 내 앞에 설지 알 수 없다.
3. 승차 거부: 단거리라고 하면 그냥 가버린다.
4. 경로/요금: 미터기만 쳐다보며 빙 돌아가는 건 아닌지 불안하다.
5. 결제/분실: 내릴 때 현금/카드를 찾느라 허둥대로, 물건을 놓고 내리면 찾기 막막하다.



카카오 T 이전의 세계

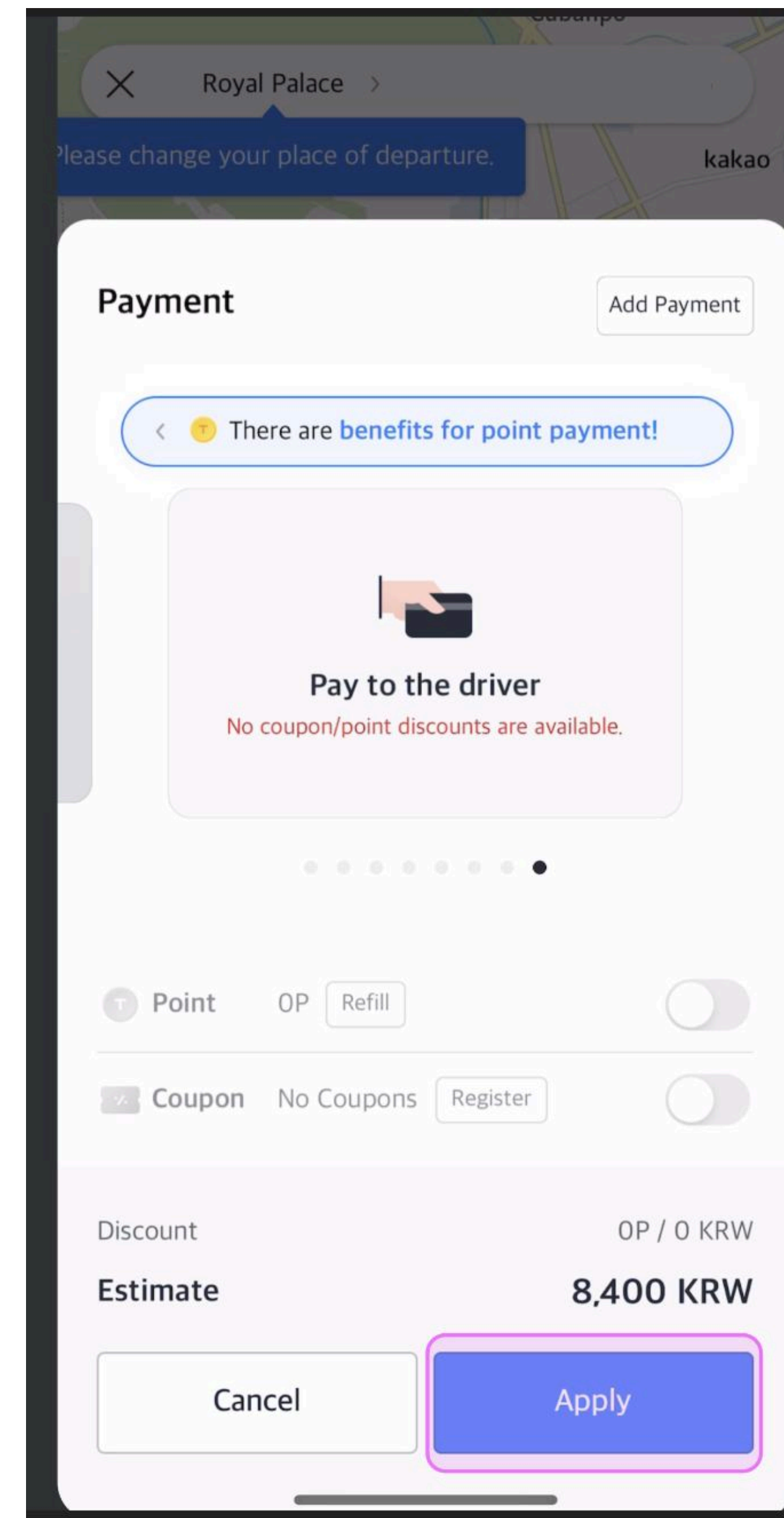
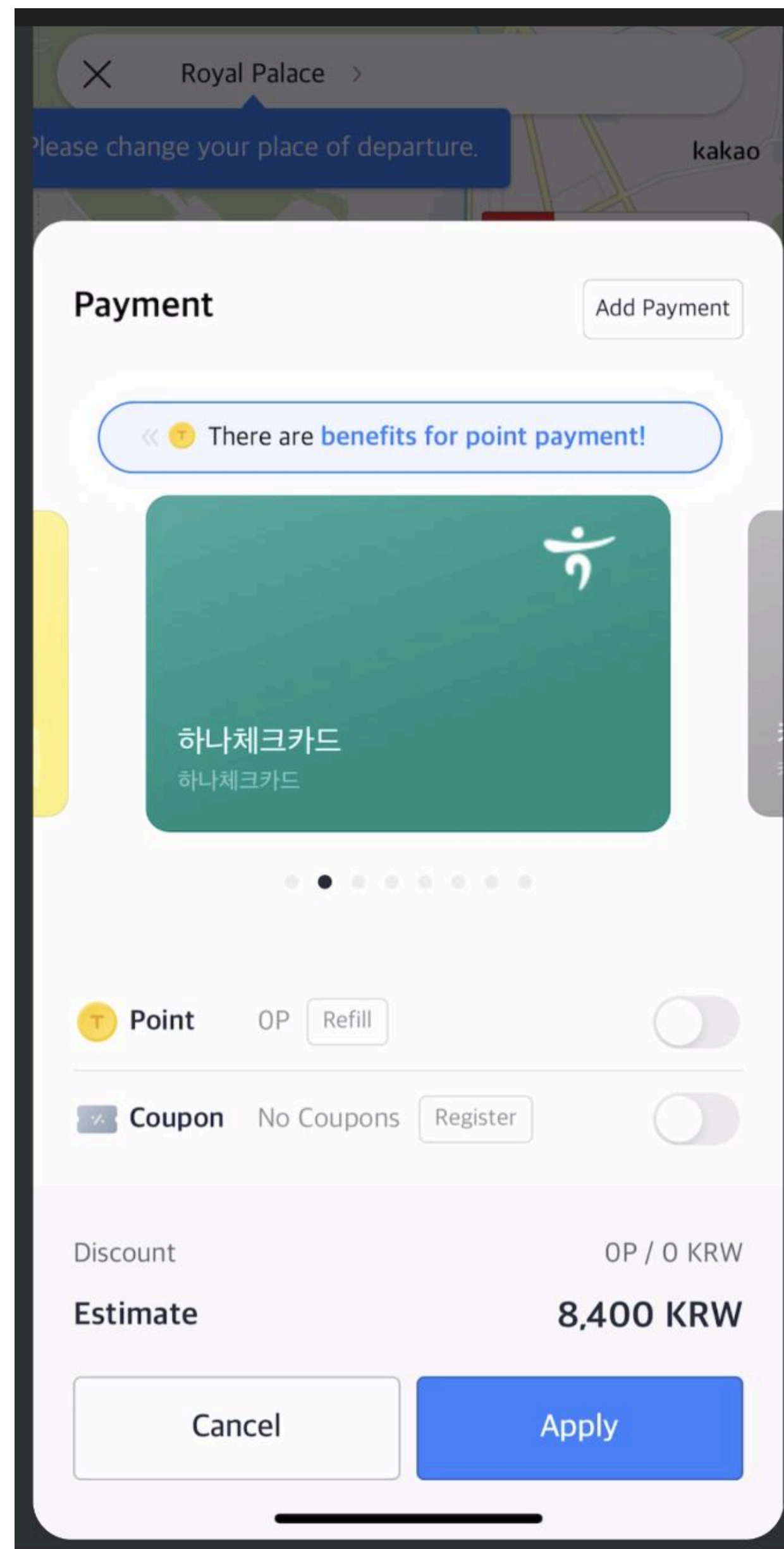
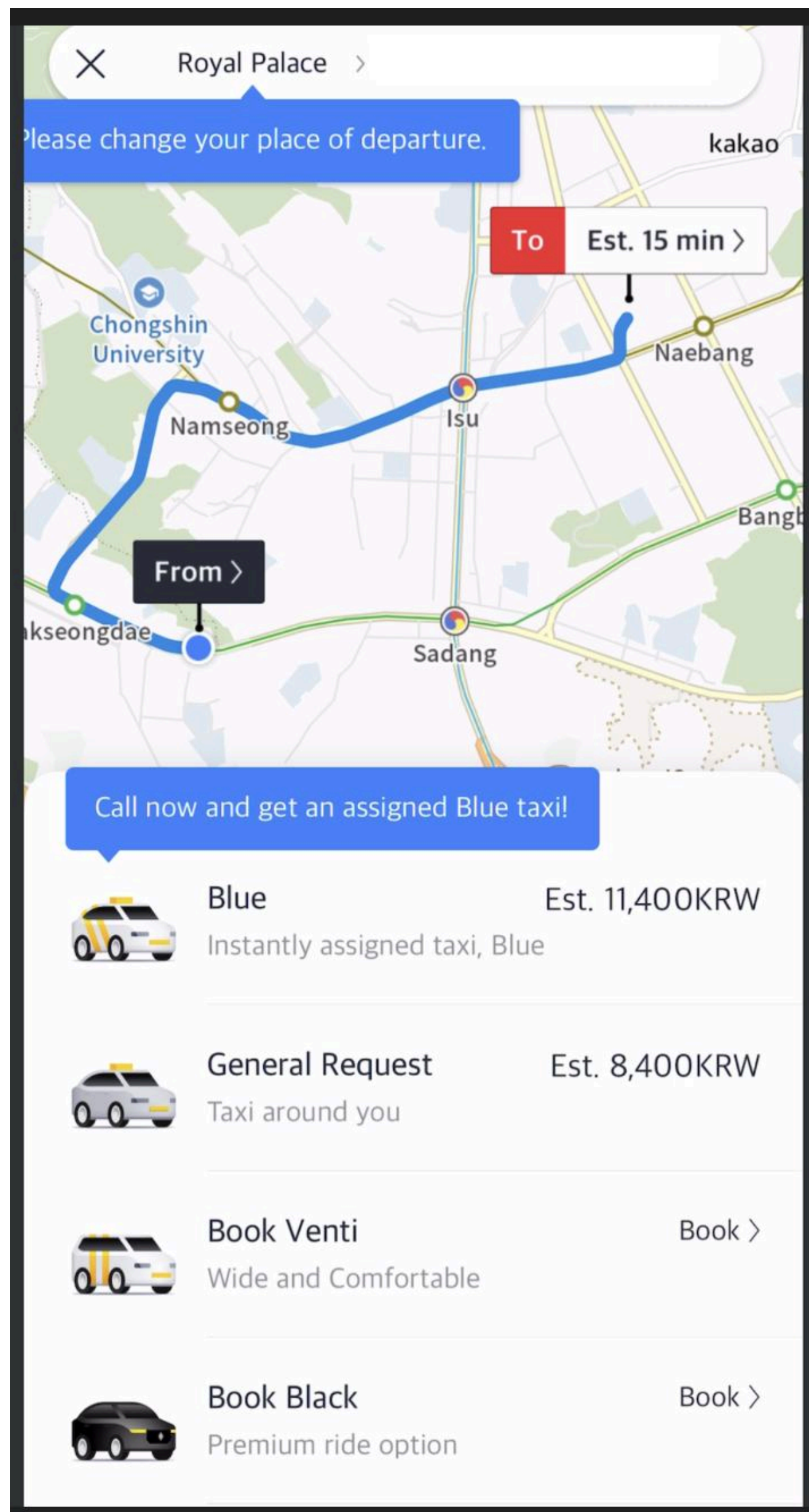


As-Is **(택시 잡기의 고통)**

1. 기다림: 길가에 나가 하염없이 손을 흔든다
2. 불확실성: 택시가 올지, 언제 올지, 내 앞에 설지 알 수 없다.
3. 승차 거부: 단거리라고 하면 그냥 가버린다.
4. 경로/요금: 미터기만 쳐다보며 빙 돌아가는 건 아닌지 불안하다.
5. 결제/분실: 내릴 때 현금/카드를 찾느라 허둥대로, 물건을 놓고 내리면 찾기 막막하다.

To-Be **(카카오T가 만든 택시 경험)**

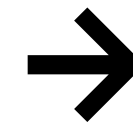
1. 호출: 실내에서 편안하게 버튼 하나로 택시를 부른다.
2. 투명성: 택시 위치, 도착 예정 시간, 기사님 정보를 실시간으로 확인한다.
3. 매칭: 목적지가 미리 공유되어 승차 거부 걱정이 없다.
4. 경로/요금: 예상 경로와 요금을 미리 보고 안심한다.
5. 결제/분실: 자동 결제로 그냥 내리고, 이동 기록이 남아 분실물을 찾기도 쉽다.



As Is - To Be

고객의 고통 (Pain Points)

택시를 잡는 모든 과정이 불확실하고, 불안하고, 불편하다.



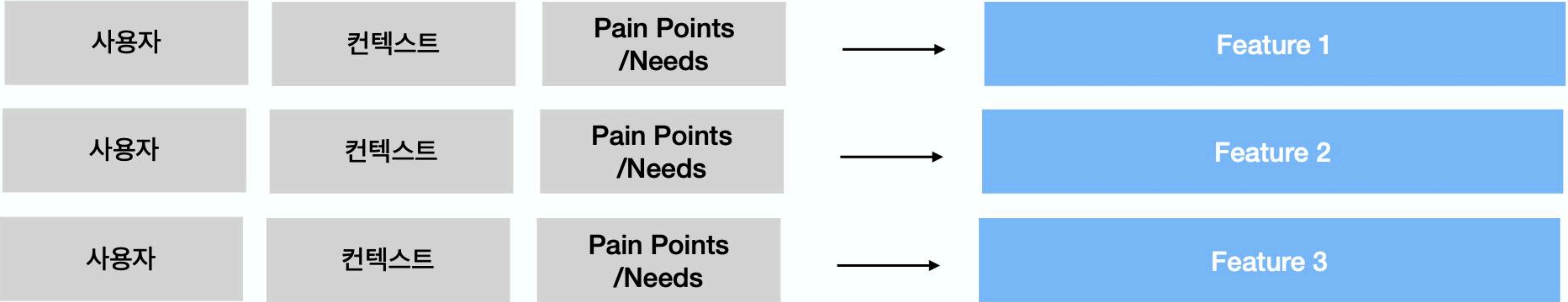
우리의 핵심 가치 (Value Proposition)

언제 어디서든, 가장 확실하고 편안하게 택시를 이용하게 한다.

택시 앱 → 택시 잡는데 고통을 겪는 모든 사람들을 위해,
이동 전 과정을 투명하고 편리하게 만들어주는 서비스

AS IS

TO BE



AS IS

20대 여성

온라인에서 옷
검색 시

사진으로는 사이즈나, 컬러, 재질
등을 알기 어렵다.



20대 여성

옷을 고를 때

내가 좋아하고, 나에게 어울리는
옷을 찾기 어렵다



셀러

물건을 팔 때

고객들이 어디서 들어오고 어떤 때
물건을 사는지 알고 싶다.



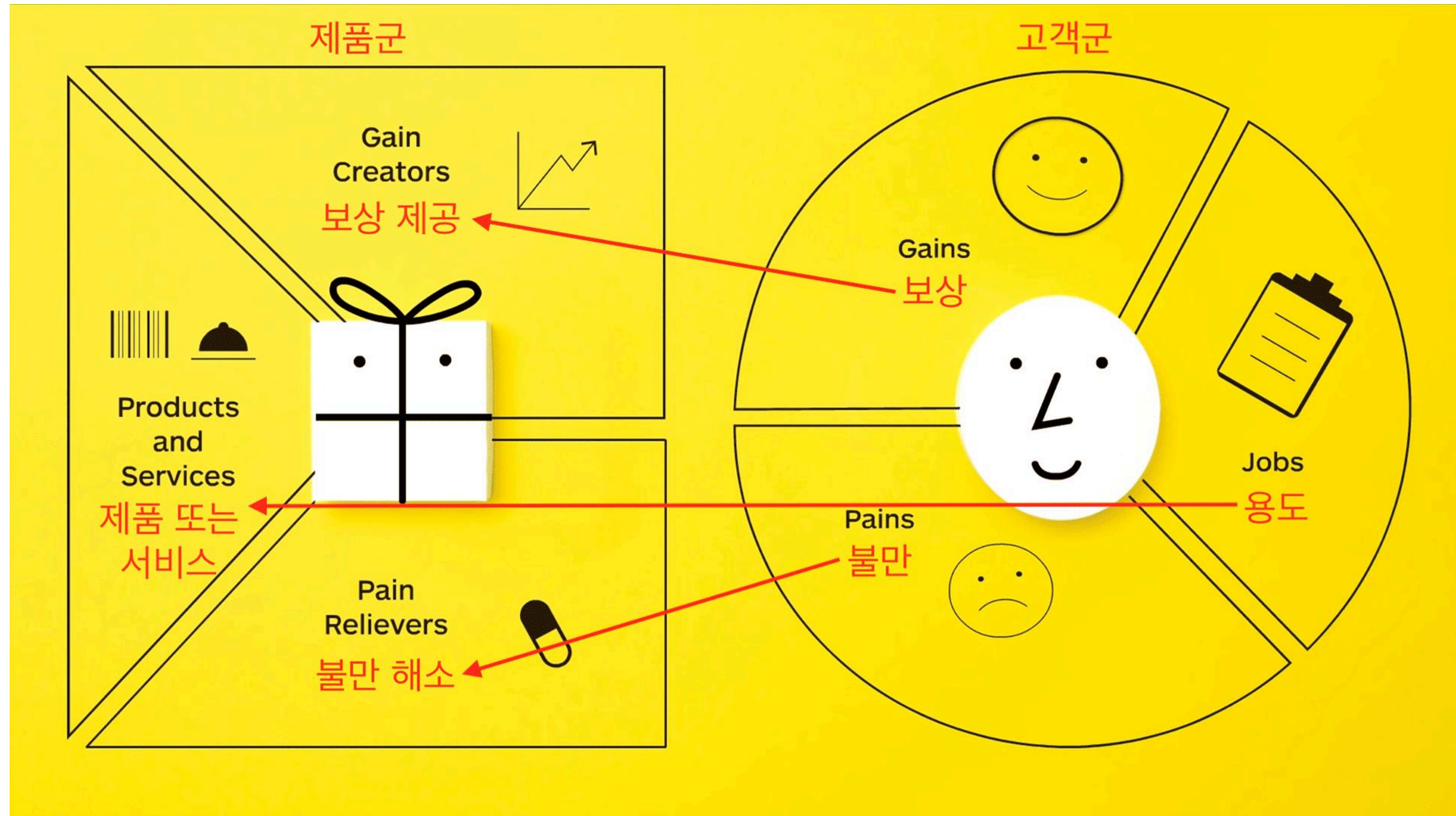
TO BE

라이브 방송을 통해 실제 제품을 확인하고 구매

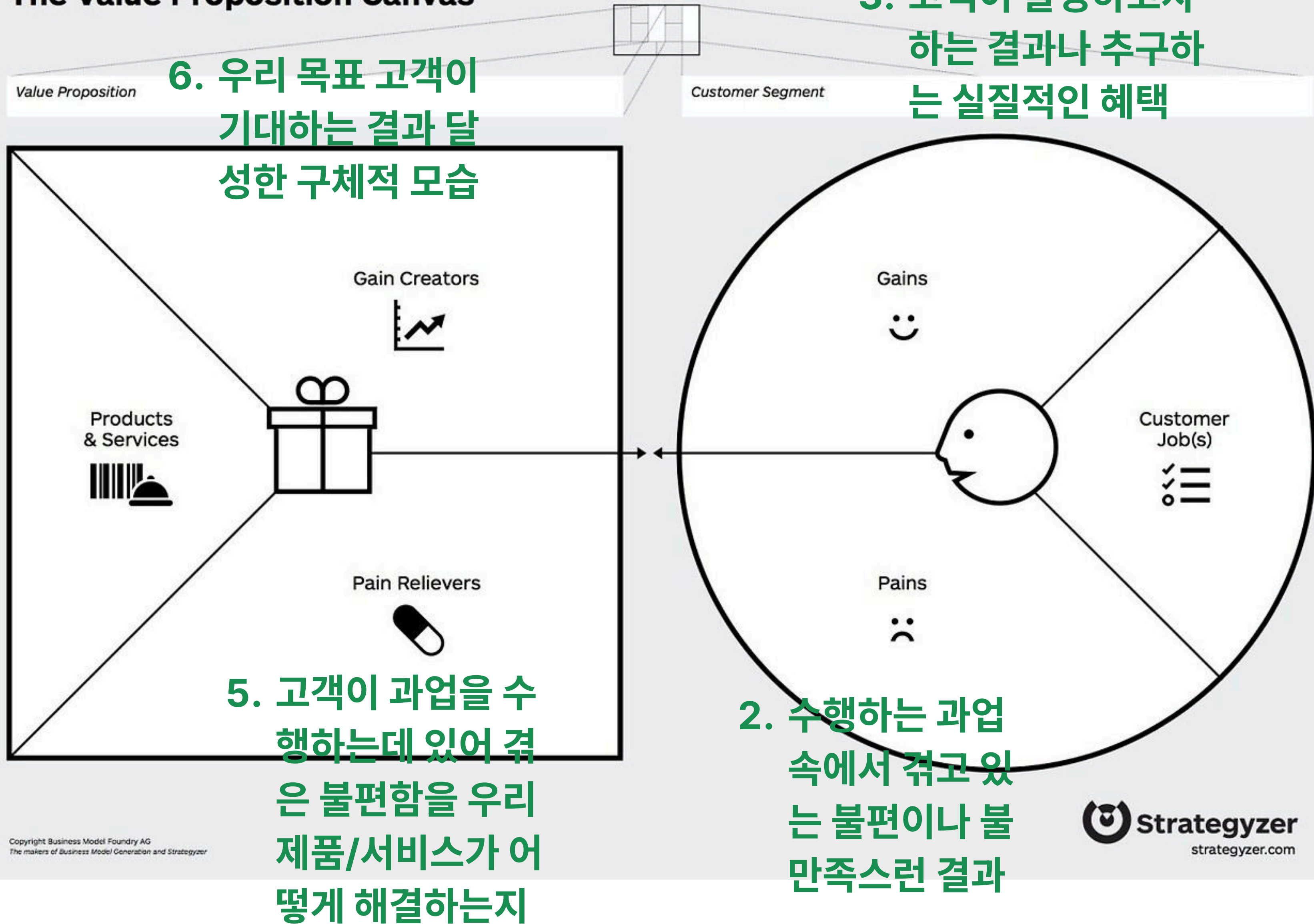
사용자의 행동 기반으로 원하는 제품/영상을
추천해준다.

방송을 보는 유저들의 행동 패턴, 유입 경로, 키워드,
영상에서의 피크 등에 대한 인사이트 제공

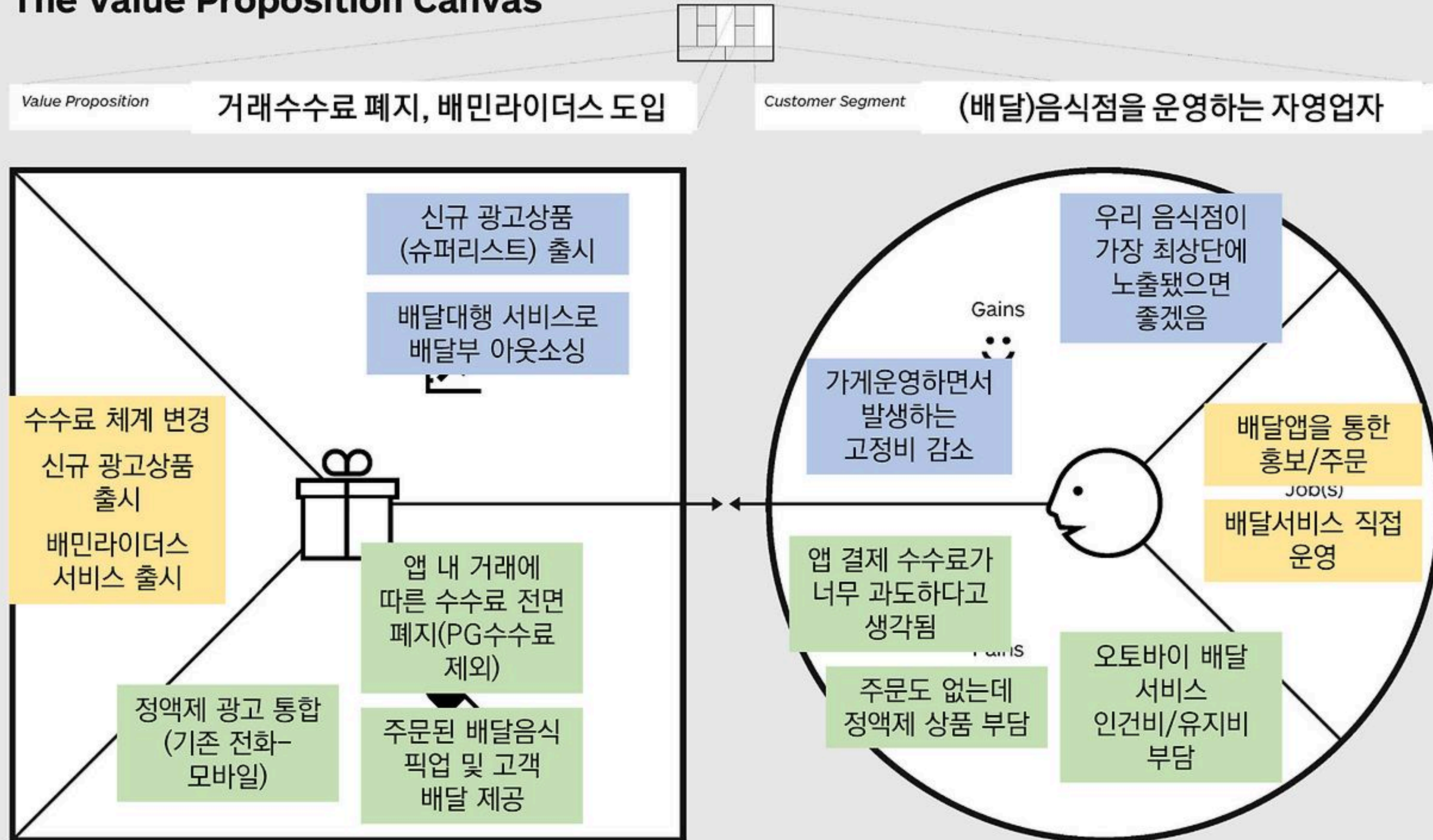
가치 제안 캔버스 (Value Proposition Canvas)



The Value Proposition Canvas



The Value Proposition Canvas



MVP / PoC

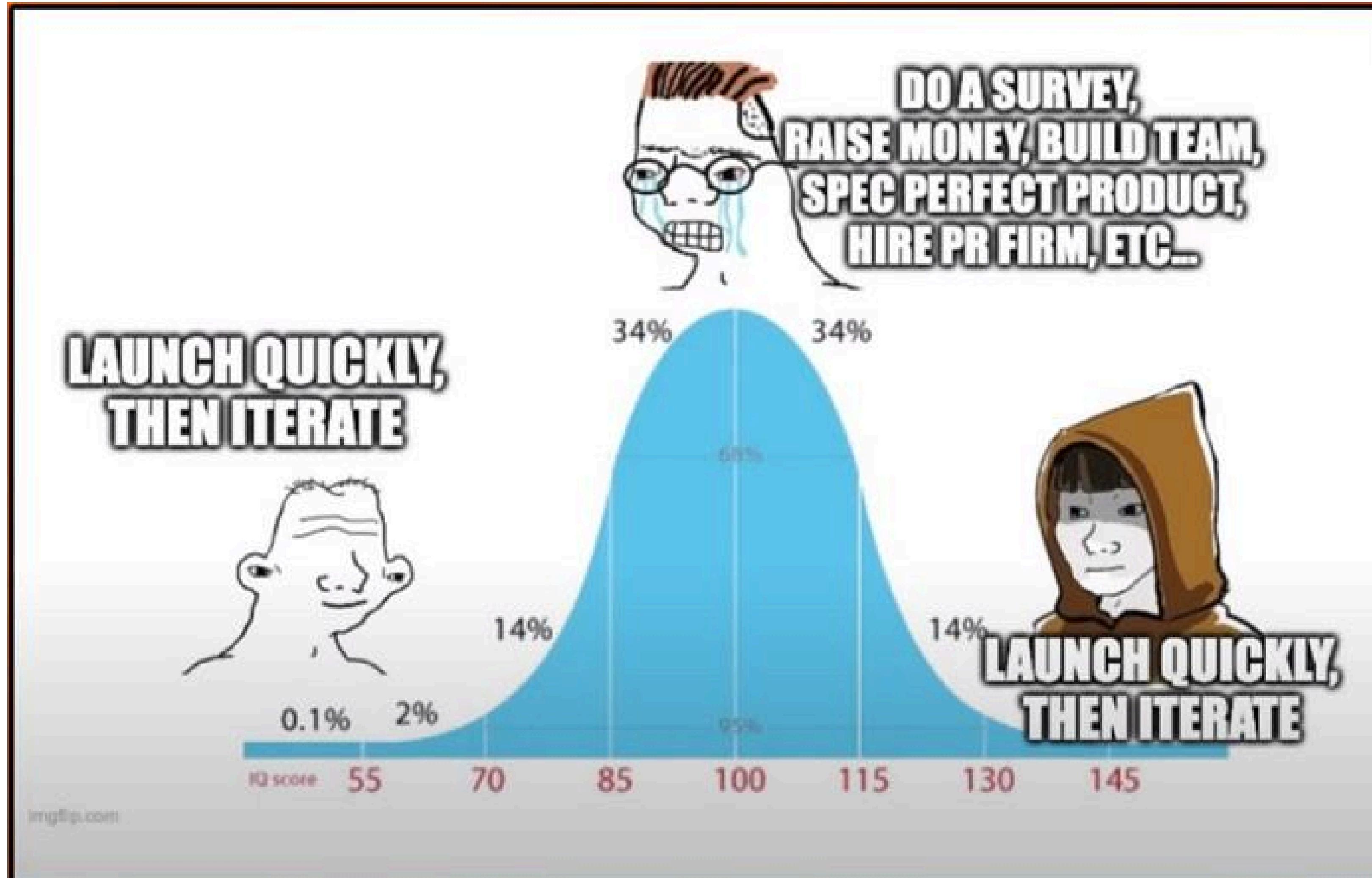
Minimum Viable Product / Proof of Concept

MVP

서비스 만들기가 어려운 이유

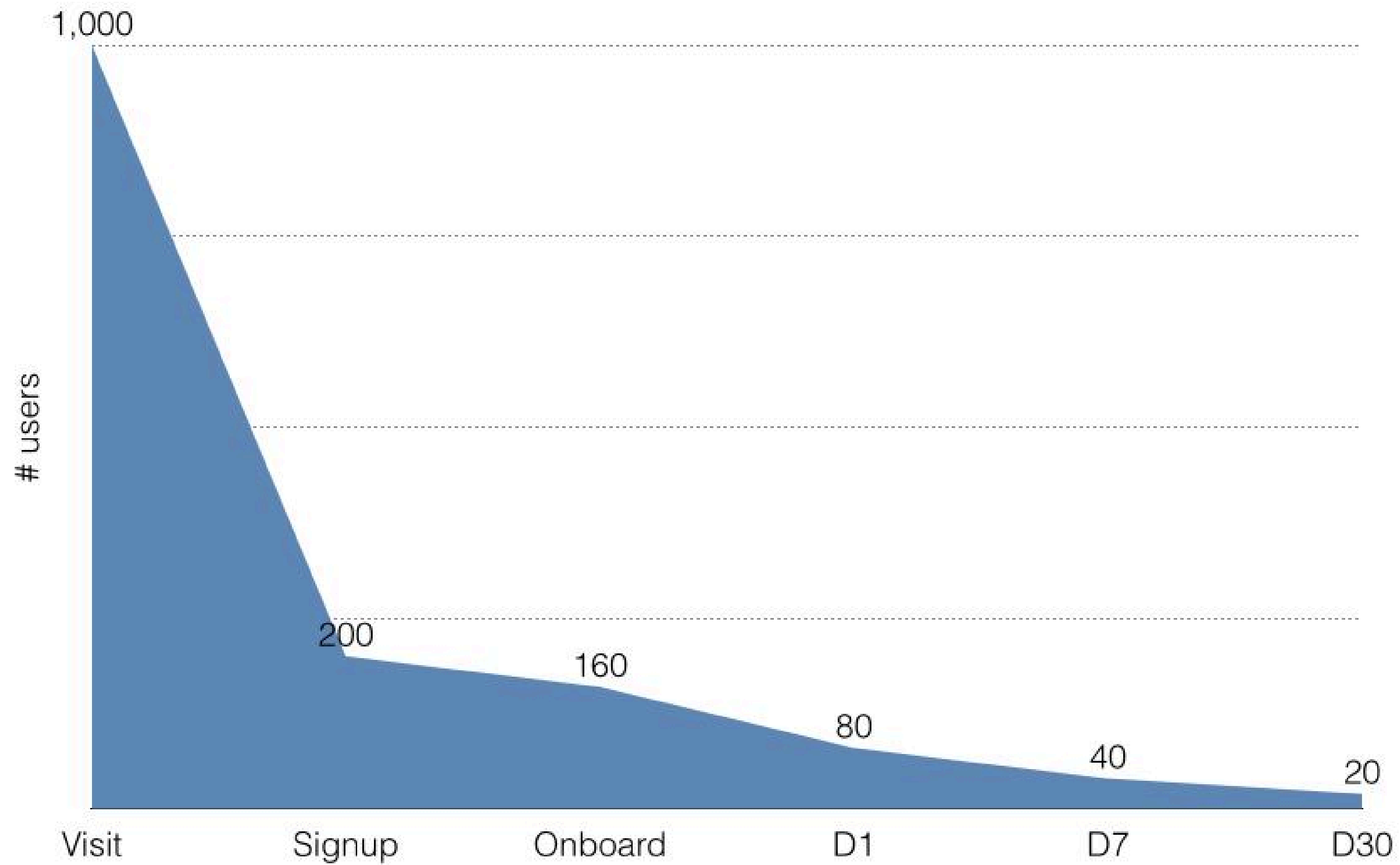


MVP를 만드는 이유



Q: 그래서 제품 어디까지 만들어야해요?

A: 우리의 핵심 가설을 검증할 수 있는 최소한까지요.



기능(Feature)이 추가될 때마다
사용자 활성도도 대폭락하는 진실

MVP 정의

Minimum Feature Product (최소 기능 제품) ✕

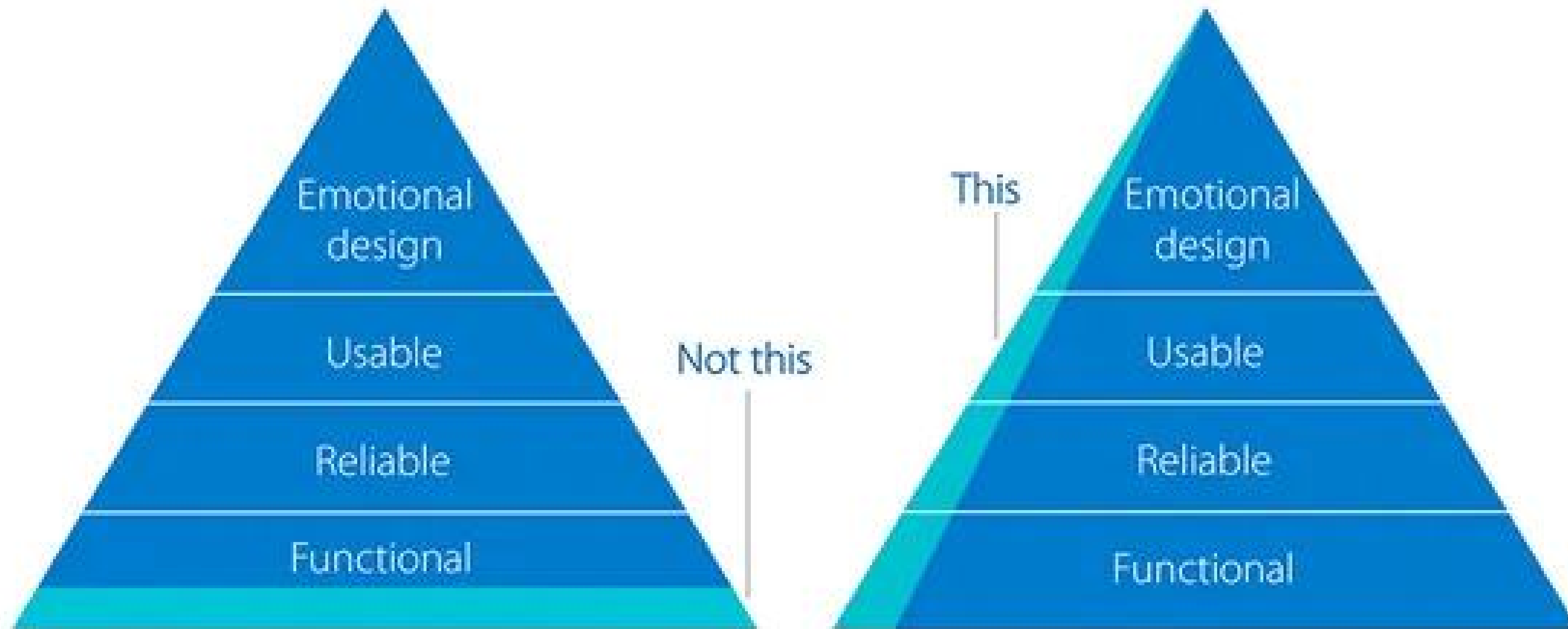
Minimum ***Viable*** Product (최소 기능 실행 가능 제품) ○

MVP 정의

Minimum Feature Product (최소 기능 제품) ✕
Minimum ***Viable*** Product (최소 기능 실행 가능 제품) ○

***Viable** 뜻: 최소한의 기능만으로도, 사용자가 우리 제품의 “핵심 가치”를 온전히 느끼고, 유의미한 “피드백”을 줄 수 있을만큼의 완성된 경험

Minimum Viable Product



@jopas

September 2014 | With compliments to Aaron Walter

***MVP* != 덜 만든 제품**
***MVP* = 작지만 완결된 경험**



완성된 접시 1개 → 코스 요리 전체



HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4



ALSO HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4



HOW TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

PoC

AI 프로덕트는 기술이 곧 제품이다

Why?

1. 모델의 지능 자체가 핵심 가치임

모델이 문제를 못 풀면 제품 자체가 성립하지 않음

2. 인터페이스가 단순함

AI 프로덕트는 UI가 단순해 PoC 자체가 MVP

구분	PoC (Proof of Concept)	MVP (Minimum Viable Product)
핵심 질문	"이게 기술적으로 가능한가?"	"이게 사용자에게 가치가 있는가?"
주요 목적	기술적 타당성 검증 (Feasibility)	시장/비즈니스 가치 검증 (Viability)
대상	내부 이해관계자, 의사결정권자	초기 사용자 (Early Adopters)
결과물	프로토타입 혹은 코드 조각	실제로 작동하고 배포된 최소 기능 제품
성공 기준	모델이 정답을 맞히는가? (Accuracy)	사용자가 이 서비스를 계속 쓰는가? (Retention)

MVP vs MLP vs MMP



이재철 Jaechul Lee · 1촌

Growth | AI | Marketer | Data | Lovable ...

[내 블로그 보기](#)

3시간 · 수정됨 ·



[바이브 코딩으로 제품 만드는 사람들의 딜레마]

어제 완벽하다고 생각했던 기능이 오늘 사용자
피드백 하나로 완전히 다시 만들어야 할 수 있습니다.

바이브 코딩으로 빠르게 만들 수 있다 보니
사용자들의 기대치도 계속 올라갑니다.

지난주에 감탄했던 기능이
이번주엔 당연한 게 되어버립니다.

빠르게 만들 수 있으니까 계속 개선하고 싶어지고,
사용자들도 더 빠른 업데이트를 기대하게 됩니다.
그래서 오히려 '완성'이라는 기준점을
잡기가 더 어려워진 것 같습니다.

그렇수록 제품이 왜 필요한지 본질에 돌아가야합니다
그리고 무엇보다 제품을 설득하는
마케팅이 중요해지는 것 같습니다.

이게 바이브 코딩 시대의 새로운 현실인 것 같습니다.

AI 나오면서 MVP에 대한 사람들 기대치가 매우 높아짐

그래서 나온 개념이 **MLP, MMP**

MLP = MVP + 사랑스러운 경험 (Lovable Experience)

단순히 작동하는 것을 넘어,
사용자가 처음 제품을 접했을 때
긍정적인 감정과 매력을 느끼도록
디자인과 사용자 경험(UX)에 중점

MLP (Minimum Lovable Product)

최소 사랑스런 제품 주요 목적

- **초기 팬 확보:** 긍정적 첫 인상 → 충성도 높은 초기 사용자 확보
- **브랜드 이미지 구축:** 긍정적 사용자 경험 → 좋은 브랜드 이미지
- **자발적 확산 유도:** 사용자들 만족 → 자발적 제품 추천 및 공유

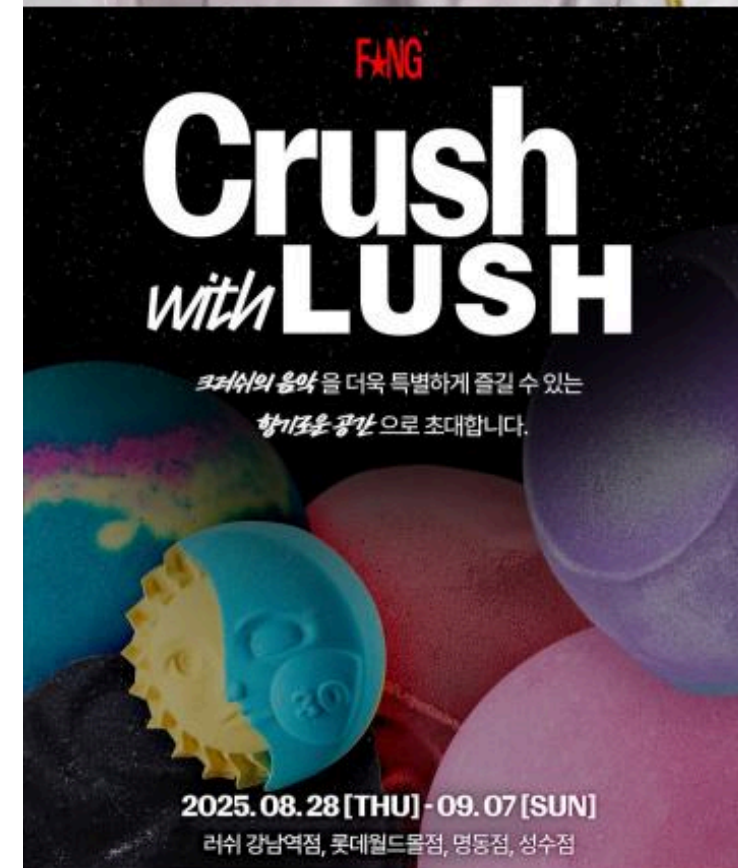
기능적으로는 최소한이지만,
사용자가 사랑에 빠질 만한 감성적 만족감 제공



9월 첫째 주 성수동 팝업

POP UP CALENDAR

매주 목요일, 핫한 성수 팝업을 큐레이션합니다.



MMP

(Minimum Marketable Product)

고객이 기꺼이 돈을 지불하고 구매할 만한
“최소한의 시장성(Marketability)”를 갖춘 제품
가설 검증을 넘어, 실제 비즈니스로서 수익을 창출하고
시장에 진입하는 것을 목표로 함

MMP (Mimum Marketable Product)

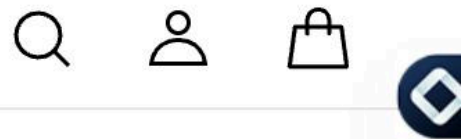
최소 판매 가능 제품 주요 목적

- **수익 창출:** 제품 판매를 통해 즉각적인 투자 수익(ROI) 목표
- **시장 출시:** 특정 고객층을 대상으로 하는 공식적인 첫 버전 제품
- **고객 확보 및 유지:** 경쟁 제품 대비 명확한 가치 제공하여 고객 유지

**고객이 느끼는 불편함을 해결해주고, 그 가치에 대해
비용을 지불할 의향이 있는 최소한의 기능 집합**

캘리쌤의 브이로그 영어

BEST



홈 / Best



(!!9월 반값 할인!)캘리쌤 브이로그 영어 전자책

상품요약정보	캘리쌤 브이로그 영어 전자책(1,000페이지 PDF, 52개 MP3파일, 편 영상링크 제공) 원어민 영어 회화 정복을 위한 완벽 공부법
소비자가	99,000원
판매가	39,000원
국내·해외배송	국내배송
배송방법	택배
배송비	무료

학습지 옵션 ☐ 31~52편(50% 할인🔥) ☐ 1~52편(9월 1만원 할인) ☐ 11~52편
[필수] 옵션을 선택해 주세요

TOTAL (QUANTITY) 0 (0개)

구매하기

장바구니

MVP vs MLP vs MMP

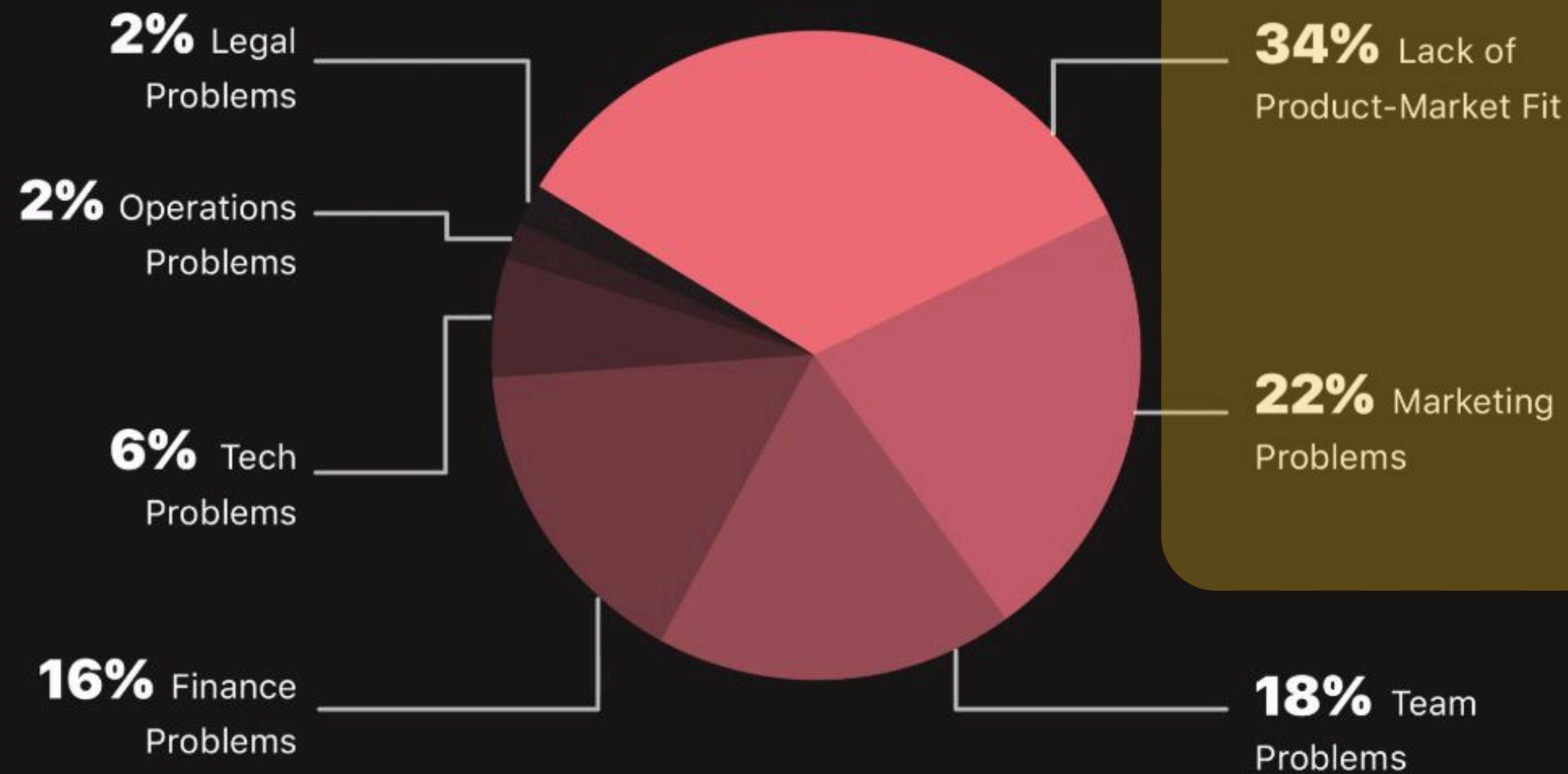
구분	MVP (최소 기능 제품)	MLP (최소 사랑스러운 제품)	MMP (최소 판매 가능 제품)
관점	Viable (실행 가능한가?)	Lovable (사랑스러운가?)	Marketable (판매 가능한가?)
주요 목표	가설 검증 및 학습	초기 팬 확보 및 경험	수익 창출 및 시장 출시
포함 기능	문제를 해결하는 '핵심' 기능	핵심 기능 + '매력적인' 사용자 경험	고객이 돈 낼 만한 '가치 있는' 기능
대상 고객	혁신가, 얼리어답터	감성적 가치를 중시하는 초기 사용자	명확한 니즈를 가진 특정 시장 고객
성공 지표	유효한 학습량, 고객 데이터	사용자 만족도, 재사용률, 추천 지수	매출, 이익, 시장 점유율

MVP 검증

우리가 뽑은 문제에 대한 MVP가 유효한가?

스타트업이 망하는 2번째 이유 = 마케팅 제대로 안되서 아무도 안씀

Common Reasons For Startup Failure

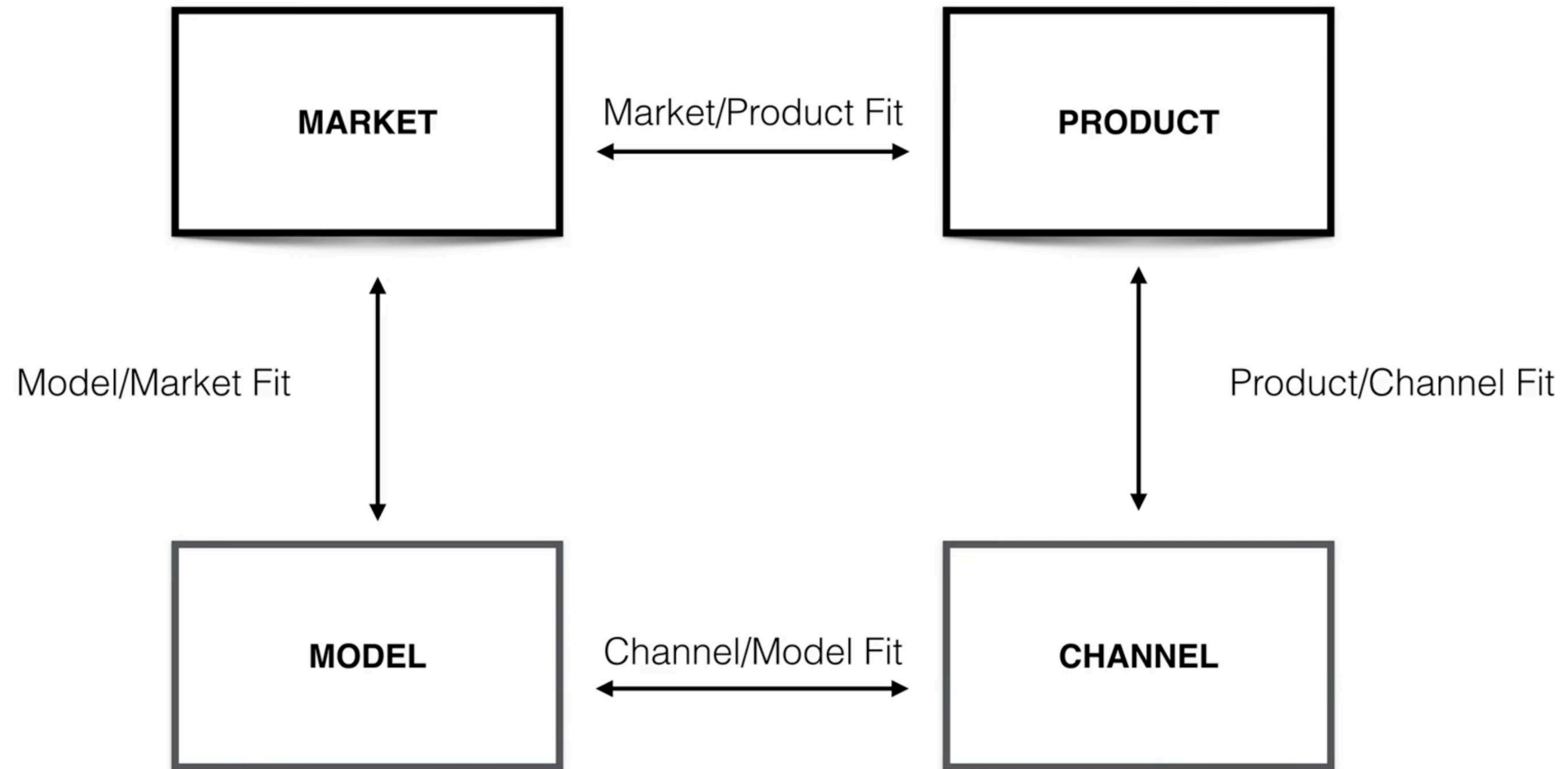


Information from 80+ failed startup interviews we've carried out.

**대부분의 스타트업은 제품 개발에
실패해서 망하는게 아니라,
고객을 얻는데 실패해서 망한다.**

Most startups fail due not to the failure of product development but due to the lack of customers. - Steve Blank

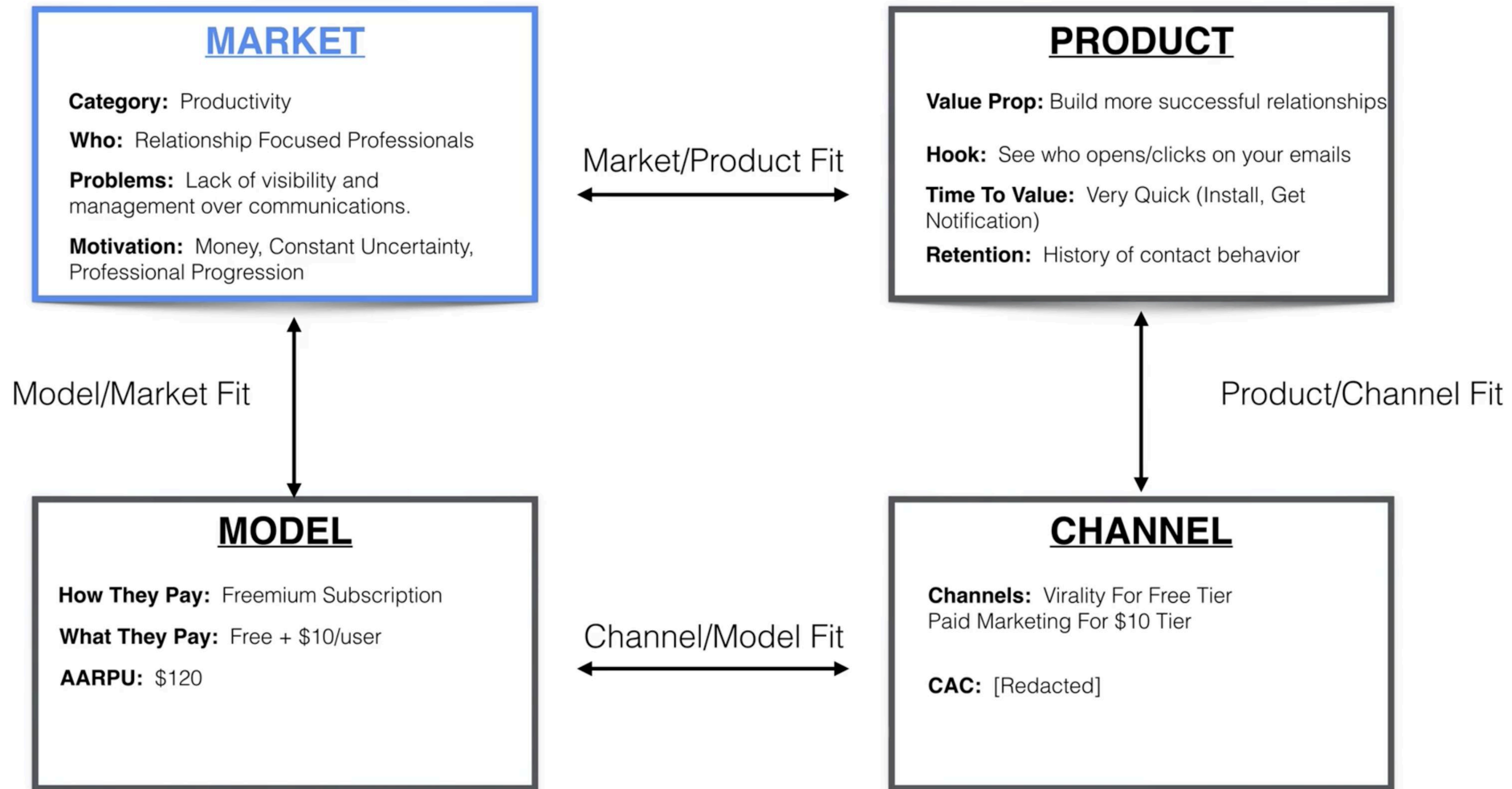
What's Needed To Build \$100M+



Copyright: Brian Balfour and Reforge Inc. <http://reforge.com>, <http://coelevate.com>.

<https://brianbalfour.com/four-fits-growth-framework>

HubSpot Sales Four Fits Canvas



What's Needed To Build \$100M+

Market <> Product Fit

We need to think about the market first, not the product.

1

Product <> Channel Fit

Products are built to tap into specific channels. You can't treat the two in silos.

2

Channel <> Model Fit

Specific channels fit with specific models.

3

Model <> Market Fit

Your model must fit with the markets pain and perceived value of your solution.

4

**“Launch early,
Learn Often”**

Paul Graham,
founder of Y Combinator

무조건 빠르게 출시하기

- 조기 출시 (신중하게 모니터링)
- 실제 사용자의 사용 패턴 관찰
- 패턴에서 문제점 빠르게 감지
- 신속한 대응 가능 → 리스크 관리

시장에서 먹고 있다는 신호(Pull)

Signs the Market is Pulling



Retention



Conversion



Referrals



Rich Feedback

시장에서 먹고 있다는 신호

1. 리텐션 우위: 유저가 스스로 돌아온다
2. 자발적 전환: 무료에서 유료로 스스로 넘어온다
3. 바이럴 추천: 전체 유입의 30% 이상이 지인 초대
4. 구체적 피드백: 단순 기능 요청이 아닌 상황 기반 요구
5. 재방문 패턴: 한 번 거절했던 고객이 몇 달 후 다시 연락

User Journey, User Stories

경험의 뼈대 설계하기

그래서 사용자는 우리 서비스를 **어떻게 경험하게** 되나요?
그래서 우리는 정확히 **어떤 기능들을** 만들어야 하나요?

유저 저니 (User Journey)

- 우리가 정의한 페르소나가 우리 서비스를 인지하는 순간부터,
- “과업(Job)”을 해결하고 만족감을 느끼기까지의 모든 접점과 감정을
- 시각화한 “경험의 지도(Experience Map)”

고객의 경험과 감정을 이해하기 위함
어떤 어려움을 겪고 있는지 찾아내기

유저 저니를 그리면서 사용자의 페인 포인트 찾기



Epic

유저 행동

유저 행동

반응/감정

Needs & PainPoints
(가설)

필요 검증 데이터

과제 영역 (기능)

Epic

유저 행동

유저 행동

반응/감정

Needs & PainPoints
(가설)

필요 검증 데이터

과제 영역 (기능)

Epic

유저 행동

유저 행동

반응/감정

Needs & PainPoints
(가설)

필요 검증 데이터

과제 영역 (기능)

“누가, 어떤 문제를, 왜 겪고 있는가”

고객이 불편해하는 구체적인 상황/시점
= 문제의 문맥 (Customer Context)

Market Promise Framework



Connect
market **Gap**

Key Unique Attributes	Value of Unique Attributes
What is the problem? (that no one else is solving)	
Who has this problem?	
What do they say when the problem arises?	
Trigger in Time: (WHEN does this happen to them?)	
What issues is this problem causing?	
Why are competitors getting this wrong?	
What unique Functionality (do ONLY we have that can solve this?)	
What does this enable?	
Put it all together! 👉	

To product
solution

EXAMPLE

Market Promises: Fast social posts

Canva



Connect
market Gap



To product
solution

Key Unique Attributes	Value of Unique Attributes
What is the problem? (that no one else is solving)	It's incredibly difficult (or costly) to product quality social media content daily.
Who has this problem?	Marketers, social media managers, influencers.
What do they say when the problem arises?	<i>"We have nothing to post next week, and our designer freelancer is costing us a fortune. The only other designer we have is on leave! This is killing us"</i>
Trigger in Time: (WHEN does this happen to them?)	When they need to schedule next week's social posts, or when their posts needs something edited.
What issues is this problem causing?	Delayed content, low quality content, a lack of followers, a lack of engagement from social following – low results.
Why are competitors getting this wrong?	Making over complicated design tools that few people can use. High reliance on costly and external designers.
What unique Functionality (do ONLY we have that can solve this?)	Range of social media templates, and super simple design elements that anyone can use in minutes. The ease of use of Microsoft paint, but the results of photoshop.
What does this enable?	Quality social media content that can be produced by anyone quickly – with out any outsourcing costs.
Put it all together! 👉	Canva helps social media managers create quality design for posts in minutes, so they never miss a post, and keep their followers growing and engaged.

고객은 종종 자신의 병명을 모른채,
증상만을 이야기한다.

우리의 목표는 고객의 말 뒤에 숨은
진짜 “**핵심 문제**”를 진단하는 것

**고객의 “원함(Want)”과
시장의 “필요(Need)”를
구분해 알맹이 찾아내기**

사용자(User/Persona)들이
어떤 상황에서 (Context)
이런 문제 (Pain points / Needs)
때문에 이런 Features들이 필요하다!



아하 모먼트 (Aha Moment)

사용자가 제품의 '기능'을 단순히 인지하는 것을 넘어,
그 기능이 자신의 삶에 어떤 **"핵심적인 가치"**를 가져다주는지
정량적으로 & 정성적으로 깨닫는 순간

User Journey

우리 제품이 '왜' 필요한지 알기 위해,
사용자의 현재 '페인 포인트'를 그리는 지도
→ 잠재적인 Feature에 대한 인사이트 발견!



User Story

그 고통을 해결해 줄 '무엇(What)'을 만들기 위해,
개발팀이 실행할 수 있는 가장 작은 단위의 '할 일'

유저 스토리 (User Story)

- 개발팀이 실행할 수 있는 가장 작은 단위의 '할 일'
- Features, 혹은 더 작은 기능 단위를 사용하는 주체, 기능, 목적을 정의
- 모든 내용을 정의할 수 없기 때문에 유저 스토리로 무엇을 구현할지 합의

좋은 유저 스토리는 개발자가 **‘문제’와 ‘가치’를 명확히 이해해서,**
더 나은 해결책(기능)을 제안하고 싶게 만들어야 함.

유저 스토리는 개발자가 바로 코딩할 수 있도록 명확한 **‘기능’을 제시**
하는 **‘기능 명세서’가 아닌 “대화를 시작하기 위한 문제 정의서”**

User Story

우리가 왜 이 기능을 만들어야하는 지에 대한 목표 설정

**As a <사용자 유형/페르소나>,
I want to <행동/기능>,
so that <가치/결과>**

*앞에 나온 Job Story는 사용자의 '맥락'과 '동기'에 초점,
User Story는 사용자가 원하는 '기능'에 초점

Job Story

사용자의 '맥락'과 '동기에 초점을 맞춤

**When ~ 한 상황일때,
I want to ~하고 싶다,
So I can ~한 결과를 얻을 수 있도록**

*뒤에서 나올 User Story와 차별점: User Story는 사용자가 원하는 '기능'에 초점

구분	잘못 쓴 유저 스토리 (Bad 🖐️)
관점	'시스템' 중심 "사용자는 '저장' 버튼을 누른다." (시스템이 할 일을 설명)
'왜'의 유무	가치가 빠져있음 ... so that "다음 화면으로 넘어간다." (이건 가치가 아니라 현상)
구체성	너무 막연함 "사용자로서, 나는 편리한 기능을 원한다." (무엇을 만들어야 할지 알 수 없음)
역할	개발자를 '구현자'로 만들 "여기에 빨간색 버튼을 만들어주세요." (개발자의 고민을 막음)

잘 쓴 유저 스토리 (Good 👍)
'사용자' 중심 "나는 내 글을 잃어버리지 않고 싶다." (사용자의 목표를 설명)
명확한 가치 제시 ... so that "내가 쓴 내용을 잃어버릴까 봐 걱정하지 않아도 된다."
구체적인 상황 제시 "As a 바쁜 직장인은, 출근길 지하철에서..." (상황이 그려져서 공감하기 쉬움)
개발자를 '문제 해결사'로 만들 "사용자가 이 기능을 놓치지 않게 하고 싶어요." (더 좋은 대안을 고민하게 함)

User Story 목적

고객은 이런 상황에서 이런 걸 원하고 있구나.
이걸 해결해주면 정말 좋아하겠네.
어떻게 해결해주는게 가장 효과적일까?

User Story - 당근마켓



As a 이사 준비로 바쁜 1인 가구는,

I want to 동네에서 빠르게 가구를 팔아 없애버리고 싶다,

so that 불필요한 택배 포장이나 배송 약속의 번거로움을 피할 수 있도록.

User Story - 배달의 민족



As a 매일 저녁 메뉴가 고민되는 자취생은,

I want to 다른 사람들의 솔직한 리뷰와 별점을 먼저 확인하고 음식을 시키고 싶다.

so that 맛있는 음식을 시키는 '실패'의 위험을 줄일 수 있도록.

User Story - 인스타그램



As a 친구 결혼식에서 사진을 잔뜩 찍은 신부 친구는,

I want to 여러 장의 사진을 한 번에 올려 참석했던 신랑 신부와 손님들이 볼 수 있도록 했으면 좋겠다.

so that 일일이 사진을 전송하지 않아도 되고, 결혼식에서 만났던 사람들도 사진을 한 번에 볼 수 있도록.

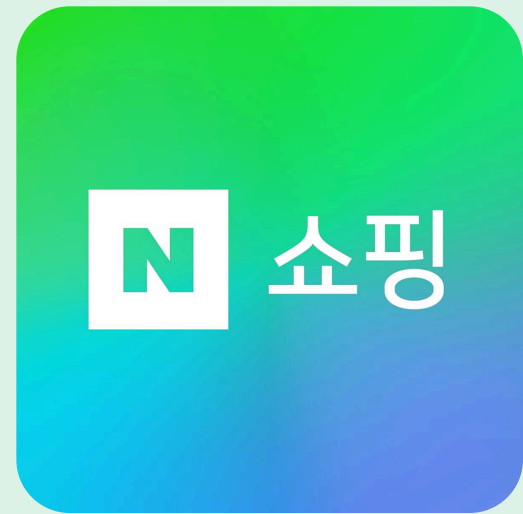
User Story - Gherkin

유저 스토리가 완료되었는지 판단하는 구체적인 “완료 조건 (Acceptance Criteria)”

**Given <주어진 상황>,
When <어떤 행동을 하면>,
Then <기대하는 결과>**

As a / I want to / So that은 “우리가 왜 이 기능을 만들어야하는지”에 대한 목표 설정
Given / When / Then은 그래서 이 기능이 구체적으로 어떻게 동작해야 성공이라고 할지

User Story - 네이버 쇼핑



Given 사용자가 [로그인]한 상태에서 장바구니에 상품이 1개 이상 담겨있을 때

When 사용자가 [결제하기]를 클릭했을 때

Then [주문 완료] 페이지로 이동하고, “주문이 성공적으로 완료되었습니다”라는 팝업창이 보여야 한다.

User Story - 무신사

A black square logo with the word "MUSINSA" in white, sans-serif, uppercase letters.

Given 사용자가 [로그인]한 상태이고, 'A 브랜드의 반팔 티셔츠' 상품 상세 페이지를 보고 있다. 그리고 상품의 하트 아이콘이 비어있다.

When 사용자가 상품 이미지 아래 하트 아이콘을 누르면,

Then 하트 아이콘이 즉시 빨간색으로 채워져야 한다.

And "찜한 상품에 추가되었습니다"라는 토스트 메시지가 화면 하단에 잠시 나타나야 한다.

And 사용자가 [마이 페이지]의 [찜한 상품] 목록으로 이동했을 때, 'A 브랜드의 반팔 티셔츠'가 목록에 추가되어 있어야 한다.

Gherkin 포맷

Title : {사용자}는 {테스크}를 수행 할 수 있다.		Medium Priority
User Story (Background)	As {사용자} I want {니즈,골} So that {이득, '##' 하기 위해서}	
Acceptance Criteria (구현에 대한 기준)	Given {Where : 주어진 상황} When { 조건 및 행동 } Then {결과}	
	* 하나의 문장에 하나의 Action을 작성하고, 추가 Action은 And와 Or을 사용하여 추가 한다.	
Notes	기타 팀 구성원들이 알아야할 사항	



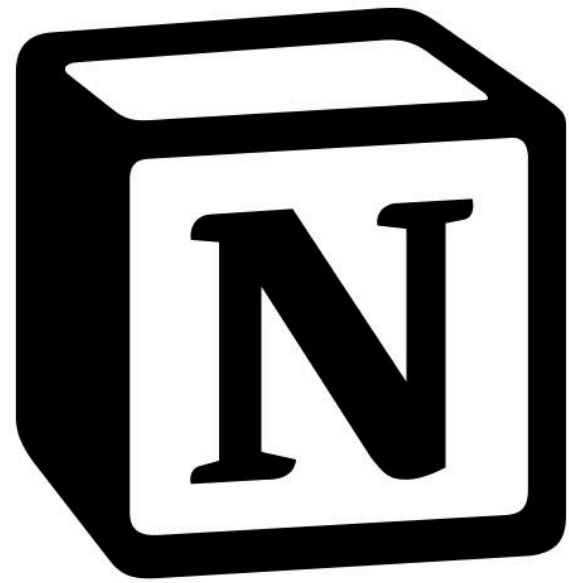
- As a 저녁 메뉴를 고르기 귀찮은 사용자,
- I want to 장바구니에 담아둔 메뉴를 바로 주문하고 싶다.
- so that 매번 메뉴를 다시 검색하고 선택하는 시간을 절약할 수 있도록.

- Given 사용자가 로그인했고, “A치킨 가게”의 ‘후라이드 치킨’이 장바구니에 담겨 있다.
- When 사용자가 장바구니 아이콘을 클릭하고, [주문하기] 버튼을 누르면,
- Then [결제 화면]으로 이동해야 한다.
- And 결제 화면의 주문 내역에는 “후라이드 치킨”이 정확히 표시되어야 한다.



- As a 피드를 구경하는 사용자,
- I want to 마음에 드는 게시물에 [좋아요]를 눌러서,
- so that 게시물을 만든 친구에게 내 관심과 지지를 표현할 수 있도록.

- Given 사용자가 로그인했고, 친구의 게시물을 보고 있으나 하트 아이콘은 비어있다.
- When 사용자가 게시물 사진을 더블탭하거나, 하트 아이콘을 한 번 탭하면,
- Then 하트 아이콘이 즉시 빨간색으로 채워져야 한다.
- And [좋아요] 숫자가 1 증가해야 한다.



- As a 체계적인 정리가 필요한 사용자,
- I want to 전문가가 만들어 둔 [템플릿]을 내 페이지에 바로 적용하고 싶다.
- so that 처음부터 양식을 직접 만드는 수고 없이, 바로 내용을 채워넣도록.

- Given 사용자가 로그인했고, 새로운 빈 페이지를 만들었다.
- When 사용자가 페이지 본문에 있는 [템플릿] 버튼을 클릭하고, [회의록] 템플릿을 선택하면,
- Then 페이지에 [회의록] 템플릿의 모든 양식(제목, 참석자, 논의 내용 등)이 자동으로 생성되어야 한다.
- And 사용자는 즉시 각 항목에 타이핑을 시작할 수 있어야 한다.

잘 구현된 유저 스토리는 개발 구현 테스트할 수 있는 테스트 시나리오로 활용 가능함

1. 사용자 로그인 기능 단위 테스트 시나리오

테스트 케이스 ID	기능	테스트 조건	입력 값	기대 결과
TC-LOGIN-001	정상 로그인 시도	올바른 ID와 비밀번호를 입력한 후 로그인 버튼 클릭	ID: user123, PW: password123	"로그인 성공" 메시지가 출력되고, 사용자가 대시보드 화면으로 이동
TC-LOGIN-002	비밀번호 틀림	올바른 ID와 잘못된 비밀번호 입력 후 로그인 버튼 클릭	ID: user123, PW: wrongpassword	"비밀번호가 틀렸습니다" 오류 메시지 출력
TC-LOGIN-003	비활성화 계정 로그인 시도	비활성화된 계정으로 로그인 시도	ID: deactivated_user, PW: password	"계정이 비활성화되었습니다. 관리자에게 문의하세요" 메시지 출력
TC-LOGIN-004	비밀번호 입력 안 함	ID만 입력하고 비밀번호를 비워둔 상태로 로그인 버튼 클릭	ID: user123, PW: (empty)	"비밀번호를 입력하세요" 메시지 출력
TC-LOGIN-005	비회원 로그인 시도	가입되지 않은 ID와 비밀번호 입력 후 로그인 버튼 클릭	ID: unregistered_user, PW: anypw	"가입되지 않은 사용자입니다" 메시지 출력

2. 상품 검색 기능 단위 테스트 시나리오

테스트 케이스 ID	기능	테스트 조건	입력 값	기대 결과
TC-SEARCH-001	정상 검색	유효한 검색어 입력 후 검색 버튼 클릭	검색어: "스마트폰"	"스마트폰" 관련 상품 목록이 화면에 표시됨
TC-SEARCH-002	검색 결과 없음	존재하지 않는 상품명 입력 후 검색 버튼 클릭	검색어: "없는상품"	"검색 결과가 없습니다" 메시지 출력
TC-SEARCH-003	검색어 입력 안 함	검색창에 아무 것도 입력하지 않고 검색 버튼 클릭	검색어: (empty)	"검색어를 입력하세요" 메시지 출력
TC-SEARCH-004	특수문자 검색	검색창에 특수문자 입력 후 검색 버튼 클릭	검색어: "!@#\$\$%"	"검색 결과가 없습니다" 메시지 출력

3. 장바구니 기능 단위 테스트 시나리오

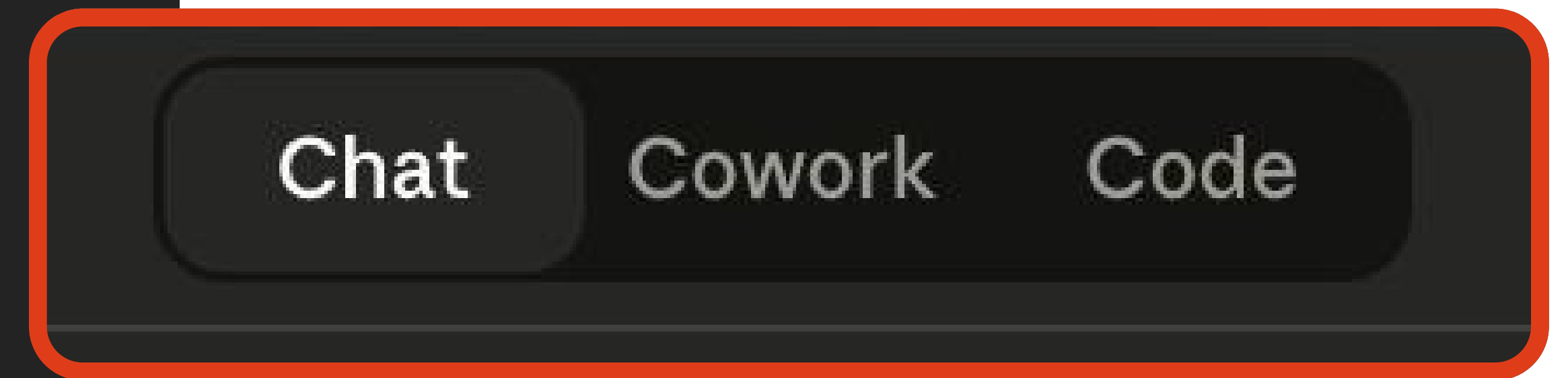
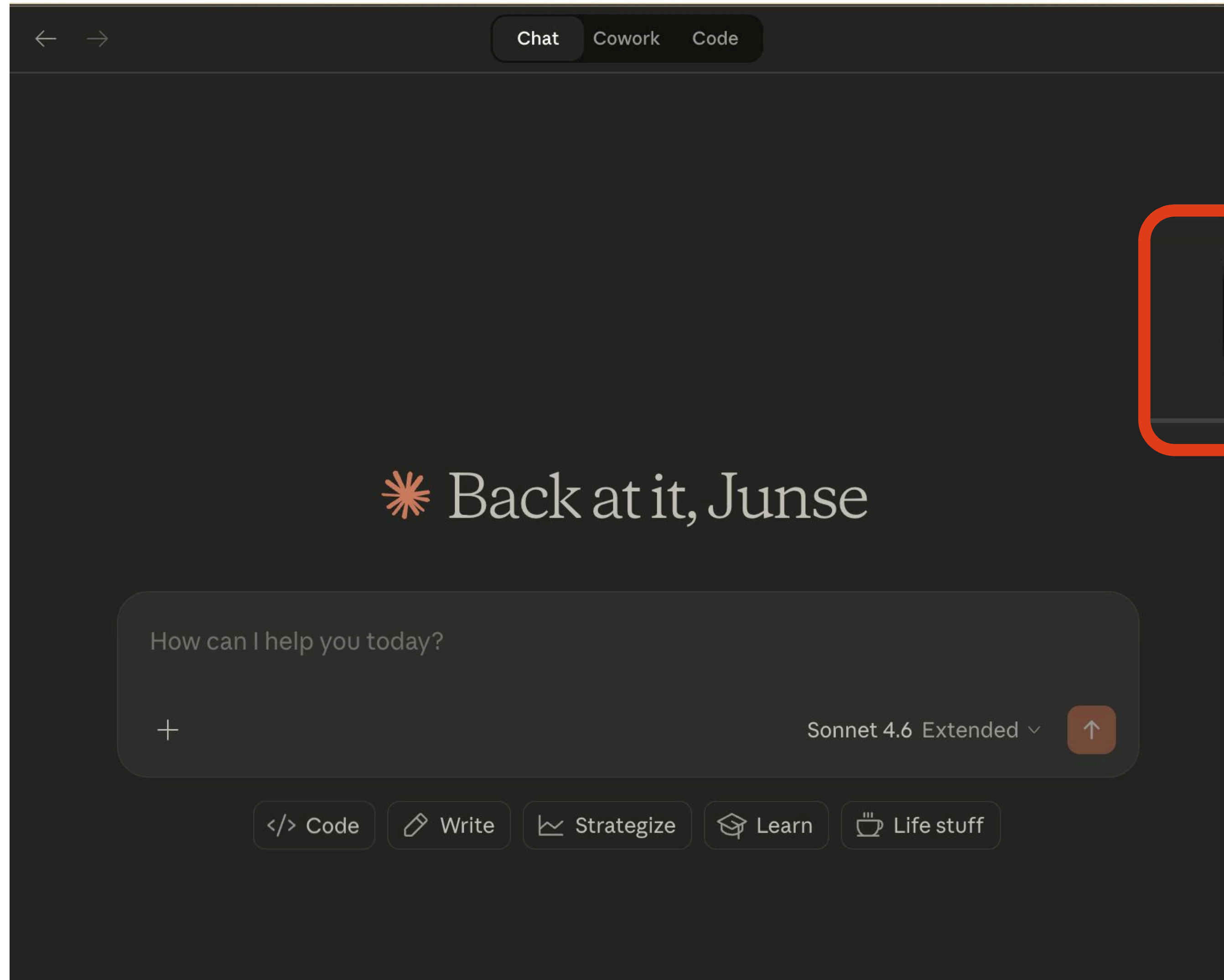
테스트 케이스 ID	기능	테스트 조건	입력 값	기대 결과
TC-CART-001	장바구니 추가	상품 상세 페이지에서 "장바구니에 담기" 버튼 클릭	상품 ID: 12345	"장바구니에 상품이 추가되었습니다" 메시지 출력, 장바구니 목록에 상품 추가
TC-CART-002	중복 상품 추가	동일 상품을 장바구니에 여러 번 추가	상품 ID: 12345	장바구니에 동일 상품의 수량 증가
TC-CART-003	품절 상품 장바구니 추가 시도	품절된 상품의 "장바구니에 담기" 버튼 클릭	상품 ID: 99999	"해당 상품은 품절 상태입니다" 메시지 출력
TC-CART-004	비회원 상태에서 장바구니 추가	비회원 상태에서 상품의 "장바구니에 담기" 버튼 클릭	상품 ID: 12345	"로그인이 필요합니다" 메시지 출력

4. 결제 기능 단위 테스트 시나리오

테스트 케이스 ID	기능	테스트 조건	입력 값	기대 결과
TC-PAY-001	정상 결제	유효한 카드 정보로 결제 버튼 클릭	카드 번호: 1111-2222-3333-4444, 유효기간: 12/24, CVC: 123	"결제가 성공적으로 완료되었습니다" 메시지 출력
TC-PAY-002	카드 정보 오류	잘못된 카드 정보로 결제 버튼 클릭	카드 번호: 0000-0000-0000-0000, 유효기간: 12/24, CVC: 123	"카드 정보가 잘못되었습니다" 메시지 출력
TC-PAY-003	잔액 부족	잔액 부족 카드로 결제 버튼 클릭	카드 번호: 1234-5678-9101-1121, 유효기간: 11/23, CVC: 321	"잔액이 부족합니다" 메시지 출력
TC-PAY-004	비회원 결제 시도	비회원 상태에서 결제 버튼 클릭	-	"로그인이 필요합니다" 메시지 출력

Building Claude Code Agents

요즘 가장 핫한 Claude Code



Chat → Cowork → Code

① 채팅(프로젝트)

- └ 인스트럭션 설정 (채널 정보, 브랜드 정체성 등)

|

② 코워크(Co-work)

- └ 프로젝트 연동 + 로컬 파일 활용

- └ 병렬 에이전트로 실무 작업 (원고 분석 + 카드 뉴스 제작)

- └ 만족스러운 워크플로우를 스킬로 저장

|

③ 클로드 코드(Claude Code)

- └ 스킬들을 플러그인으로 패키징

- └ 팀원 공유 및 반복 사용

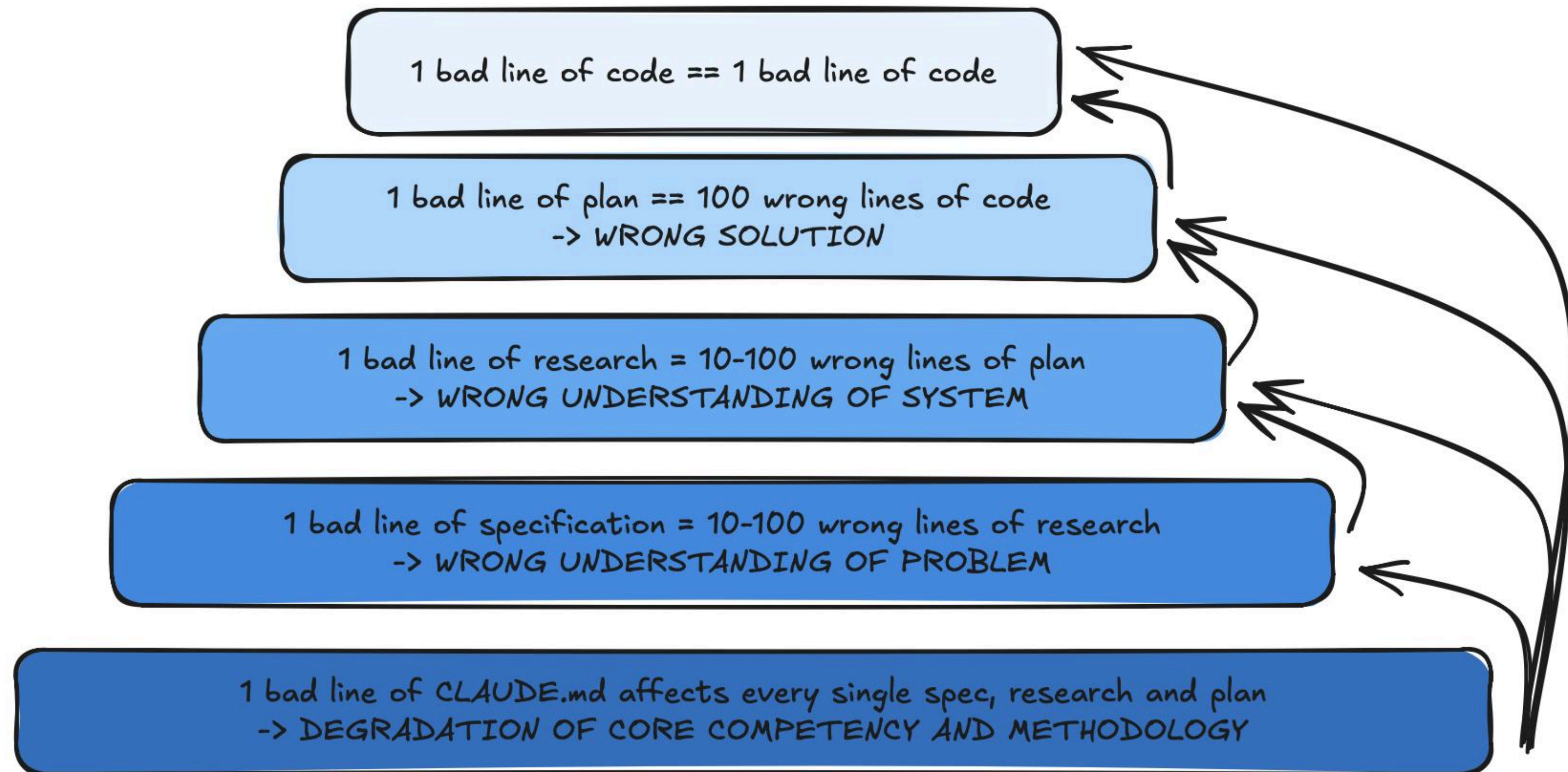
- └ 더 복잡한 작업 (홈페이지, 앱 서비스 개발 등)

CLAUDE.md = 신입 온보딩 문서

- **Why** - 프로젝트의 목적과 비즈니스 목표
- **What** - 기술 스택, 프로젝트 구조, 아키텍처
- **How** - 빌드, 테스트, 타입 체크 실행 방법

CLAUDE.md의 퀄리티 = 에이전트의 퀄리티
90%의 시간을 기획에 써야함

Hierarchy of Leverage



CLAUDE.md 포맷

```
# CLAUDE.md
```

```
## 프로젝트 개요
```

```
B2B SaaS 결제 플랫폼. Next.js 14 + Prisma + PostgreSQL.
```

```
## 빌드 & 테스트
```

- 빌드: ``bun run build``
- 테스트: ``bun test``
- 타입체크: ``bun run typecheck``

```
## 상세 문서
```

```
작업 시작 전 아래 문서 중 관련된 것을 선택해서 읽으세요:
```

- ``agent_docs/building_the_project.md`` - 빌드 파이프라인 상세
- ``agent_docs/running_tests.md`` - 테스트 전략 및 실행법
- ``agent_docs/code_conventions.md`` - 코드 스타일 가이드
- ``agent_docs/service_architecture.md`` - 마이크로서비스 구조
- ``agent_docs/database_schema.md`` - DB 모델 및 관계

프롬프트 작성 예시

에이전트가 수행할 작업

영문 논문 pdf를 번역해서 한국어 논문 pdf로 생성. 한국어 논문 pdf를 최대한 깔끔한 서식의 pdf 파일로 저장해서 아카이빙하는 목적

[작업 절차]

1. 논문 pdf가 폴더에 위치. 해당 pdf 지정
2. 원문 논문의 표, 다이어그램, 그림 등의 요소는 crop해서 추출해서 저장
3. (human review) 추출한 이미지가 누락 없이 잘 되었는지를 사람이 한번 검토
4. 원문 레이아웃을 그대로 유지하면서 영어 -> 한국어 번역 (번역은 에이전트가 직접, 양이 많으면 여러 에이전트를 병렬로 실행해서 나눠서 번역 작업)
5. 다시 취합해서 원문 레이아웃에 맞게 번역된 pdf 완성
6. (human review) 최종 번역 결과물을 사람이 검토해서 수정이 필요한 부분 지적하면 그 부분만 빠르게 수정.

프롬프트 작성 예시

에이전트가 수행할 작업

1. 유튜브 링크를 넣으면, 유튜브 대본 transcript해서 핵심 내용을 요약하고, 주요 인사이트를 도출한다. 이걸 기반으로 글을 쓸만한 글감을 도출한다
2. 여러 개의 유튜브 영상과 글감을 사용자가 지정을 해주면, 이를 기반으로 블로그 글을 완성한다.

[작업 절차]

워크플로우 1 : Youtube-Curation Pipeline

1. (human review) 사용자가 유튜브 영상 링크를 제공한다.
2. 유튜브 대본 transcript해서 각 유튜브 영상 별 transcription 결과를 파일로 저장.
3. 핵심 내용 요약하고 주요 인사이트 도출
4. 사용자에게 이 영상을 기반으로 블로그로 쓸만한 글감이나 탐구해볼만한 앵글을 제안 (내 관심사 : ai, 자동화, 1인 기업, 생산성, 강의)
5. (human review) 이 중 더 탐구하고 싶은 내용을 선택
6. 글감 폴더에 내용을 저장. 글감이나 내용은 인터넷에서 별도로 리서치하는게 아니라 유튜브 영상 본문 내용에만 의거할 것.

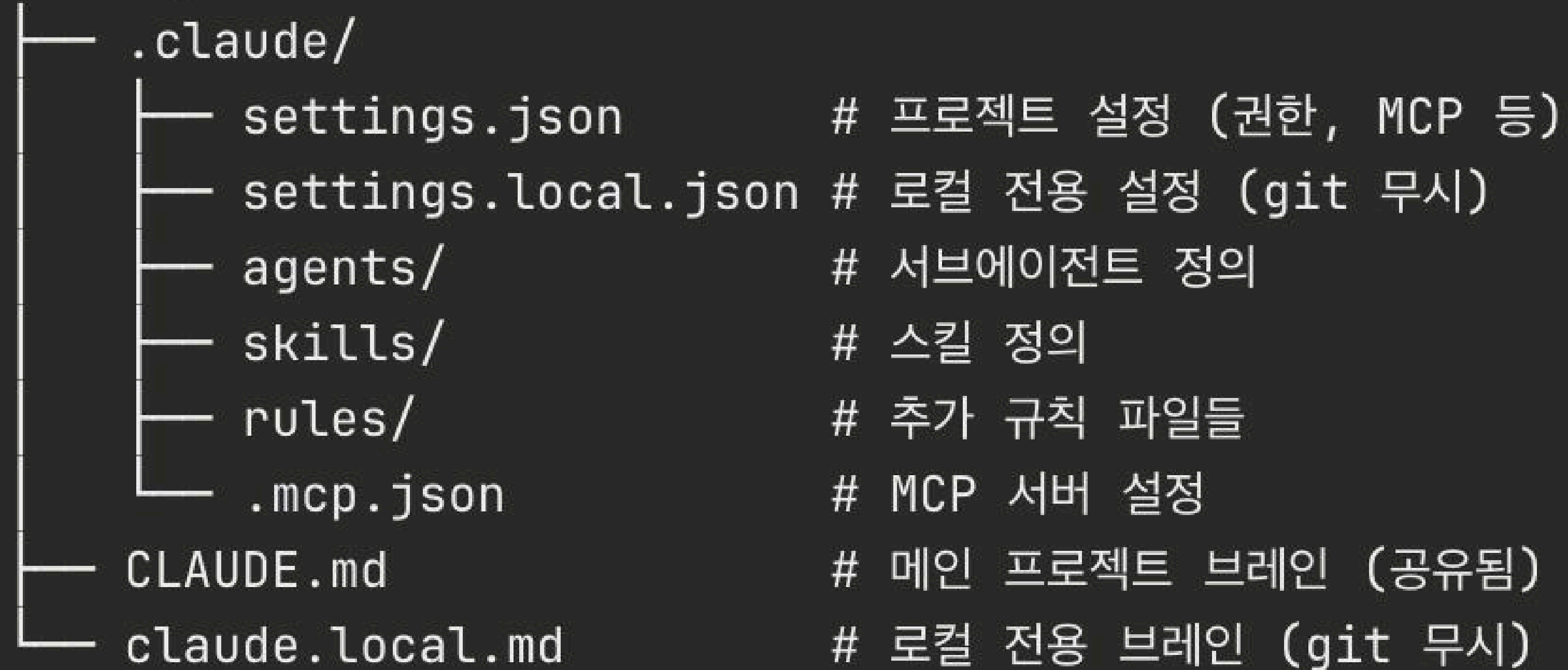
워크플로우 2:Blog-Writing Pipeline

0. Orchestrator : 전체 워크플로우를 감독 및 조율
1. Writer agent : 선별된 글감 기반 초고 작성
2. Reviewer agent : 논리를 최대한 비판적이고 냉정하게 검토. 인터넷 검색을 기반으로 팩트 체크 (공신력 있는 출처만 참고)
3. Writer agent : 피드백 반영 수정 (1~2회 루프)
- 4-1 : Style Editor : 나의 글의 스타일을 반영해서 나처럼 글쓰게 하기 위한 블로그 글쓰기 작성 스타일 가이드 문서를 관리하고 서술이나 표현 방식을 피드백.
- 4-2 : SEO Specialist : 블로그 글을 SEO 관점에서 최적화 하기 위한 피드백
5. Writer agent : 최종 반영해서 글 완성
6. (human review) 사용자가 검토하고 수정 요구사항 피드백

누가 업무 지시를 명확히하고
좋은 질문을 많이 하는가 싸움

Claude 디렉토리 구조

프로젝트/



설계 체크리스트

에이전트를 만들 때 이것들을 결정해야 해요:

항목	질문
도구	뭘 할 수 있게 할 건가? (읽기만? 수정도?)
스킬	어떤 지식을 줄 건가?
완료 조건	언제 끝났다고 판단하나?
제한 사항	하면 안 되는 건 뭔가?
출력 형식	결과를 어떤 형태로 받을 건가?
모델	sonnet(빠름/저렴) vs opus(똑똑함)?
최대 턴 수	무한 루프 방지

Agent (요리사)



Subagents (재료)

Agents

- **단일 에이전트 (기본, 권장)**
 - 워크플로우가 순차적이고 지침이 짧을 때
 - 첫 프로젝트는 단일 에이전트로 시작을 권장
- **서브에이전트 분리 (필요 시)**
 - 지침이 길어서 항상 로드하면 비효율적인 경우
 - 독립적인 작업 블록이 명확히 구분되고, 각각 다른 도메인 지식이 필요할 때

Subagents

- 메인 Claude Code 인스턴스가 독립적인 하위 에이전트를 생성
- 각 서브에이전트는 격리된 컨텍스트 윈도우를 가짐
- 메인 에이전트의 컨텍스트를 오염시키지 않음
- **서브에이전트 vs 메인 에이전트**
 - 서브에이전트: 부모가 제공한 컨텍스트만 사용
 - 토큰 비용이 추가되지만, 메인 컨텍스트 보호에 유리
 - 병렬 실행 가능 (여러 서브에이전트 동시 작업)
 - 예: code-reviewer.md, qa-tester.md

Skills

- 커스텀 명령어 + 지침 + 코드/API 호출을 캡슐화한 것
- /post처럼 슬래시 명령으로 호출하거나, 자연어로 말해도 자동 인식
- MCP보다 훨씬 토큰 효율적 (Skills > MCP)
- 재사용 가능한 워크플로우를 패키징

```
.claude/skills/스킬이름/  
├── README.md (또는 skill.md)  # Front matter + 지시사항  
└── scripts/  
    └── main.py (또는 .js 등)  # 실행 스크립트
```


Skills 디스크립션 작성 공식

- **[무엇인지]** — "OO 프로젝트의 OO 컨텍스트"
- **[언제 로드할지]** — "Auto-invoke when: OO, OO, OO를 언급할 때"
- **[언제 로드하지 말지]** — "Do NOT load for: OO, OO인 경우"
- **[소유격 사용]** — "HIS/HER/THEIR" 로 개인 스킬 오염 방지

Plugins

- Skills + Subagents + Commands를 하나로 묶어서 배포하는 단위
- 나만 쓰는게 아니라 팀에 배포해서 함께 사용할때 만들어 배포

- **Skills** → 클로드가 X를 자동으로 기억하게 하고 싶어
- **Subagents** → Y 워크플로우를 단계별로 자동화하고 싶어
- **Plugins** → 내 세팅을 다른 사람에게 공유하고 싶어

에이전트 작성 원칙

- 모든걸 담은 단일 프롬프트가 아닌 모듈형 프롬프트로 작업을 분해하라
- 지시사항의 수가 증가하면 오히려 모델 성능이 떨어짐
- 전체 스펙을 한번에 주기보다 복잡한 요구사항을 순차적이고 간단한 지시로 분해해서 준다 (한 번에 하나씩)
- **Human-in-the-loop, LLM-as-a-Judge**

에이전트 작성 원칙

근데 프롬프팅만은 아닙니다

테스트하면서 조정하는 건 프롬프트만이 아니에요:

조정 대상	예시
프롬프트	지시가 모호했네 → 더 구체적으로 수정
도구 권한	Write 줬더니 막 수정하네 → Read만 허용
스킬 조합	이 지식이 빠졌네 → 스킬 추가
모델 선택	sonnet으로 부족하네 → opus로 변경
maxTurns	무한 루프 돌아 → 턴 제한
완료 조건	중간에 끝내버리네 → 조건 명확히
구조 자체	에이전트 하나론 안 되네 → 여러 개로 분리

컨텍스트 관리

- **주요 명령어**
 - /context: 현재 컨텍스트 윈도우 상태 확인
 - /compact: 컨텍스트를 압축 (이전 대화 요약)
 - /clear: 컨텍스트 완전 초기화

The Complete Guide to Building Skills for Claude



한글 번역본: <https://www.fullstackfamily.com/@urstory/posts/13752>

각종 Skill/Agent/MCP 다운로드

<https://github.com/anthropics/skills>

<https://www.aitmpl.com/agents>

<https://skillsmp.com/>

<https://mcpservers.org/claude-skills>

Anthropic Claude Code Doc

<https://code.claude.com/docs>

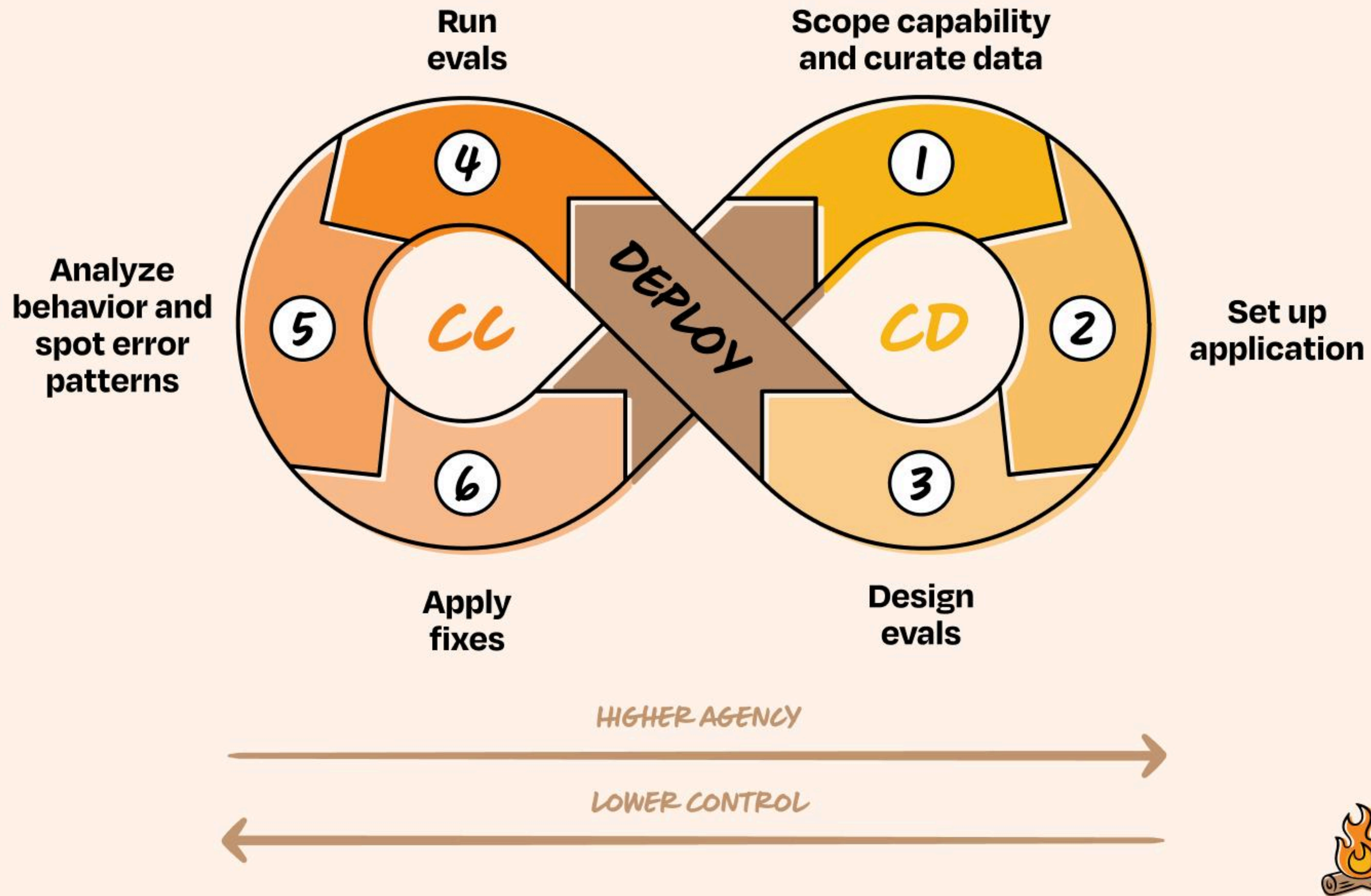
클로드 코드 70가지 꿀팁 가이드

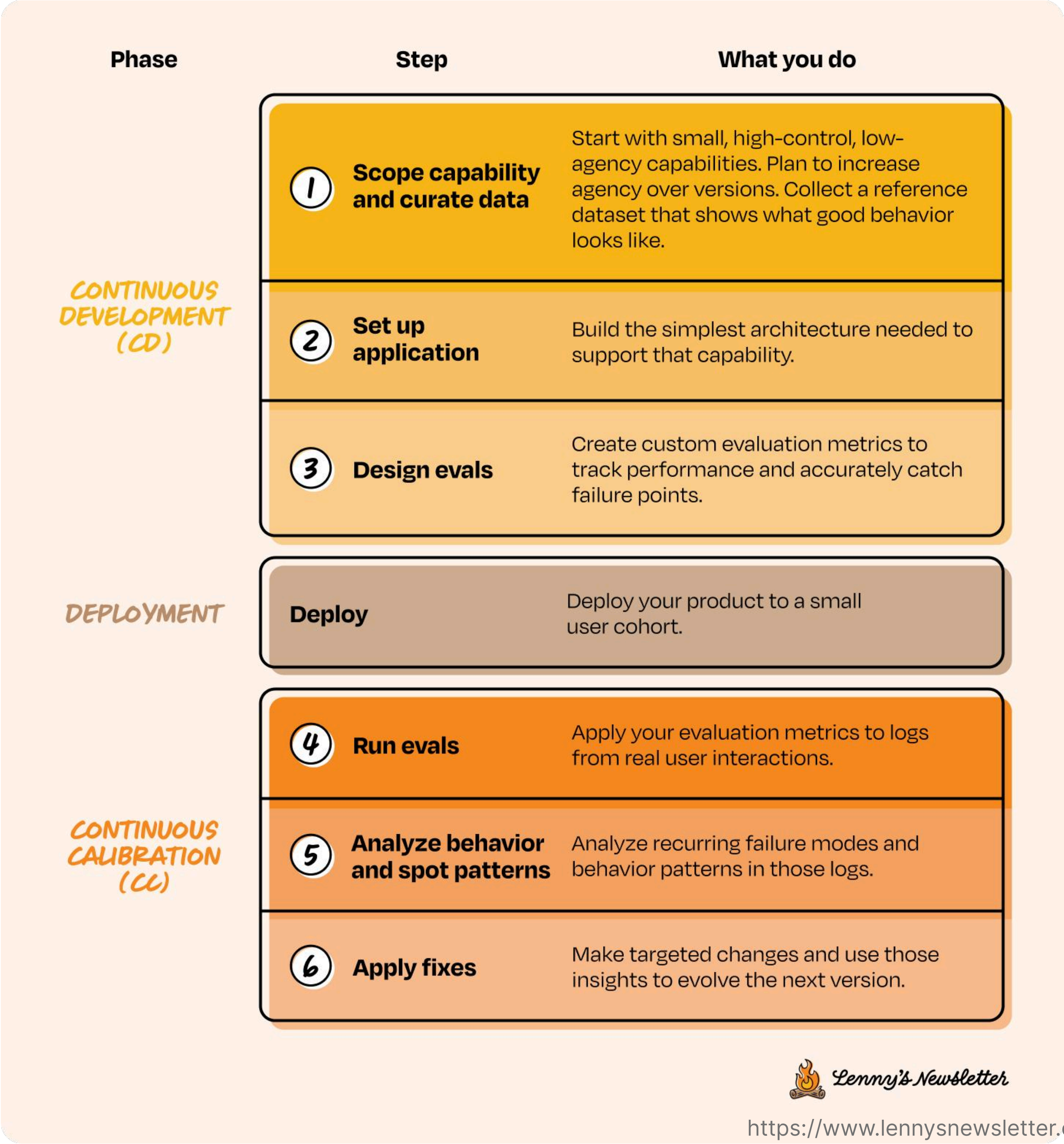
<https://drive.google.com/file/d/1x2x1T4IzTISnHGN8nd4KtMhL8iwi6QgU/view>

AI 제품 개발 주기

AI 제품 개발은 끊임없는 피드백 루프

AI 프로덕트는 일반 프로덕트와
다른 개발 사이클이 필요함





1. 문제 정의 및 AI 타당성 검토

모든 프로젝트는 “이 문제가 AI로 해결 가능한가?”에서 시작

- 비즈니스 가치 평가 = 빈도 x 절감 시간 x 인원
- 지능 유형 결정
 - 단순 요약 수준의 챗봇인지
 - 복잡한 추론과 도구 사용이 필요한 에이전트인지
- 성공 지표 수립
 - “답변 정확도 95% 이상”
 - “응답 시간 3초 이내”
 - “업무 비효율 50% 감축” 등

2. 데이터 전략 및 컨텍스트 준비

26년의 AI 성능은 모델보다 어떤 컨텍스트를 주느냐에 따라 결정

- 데이터 소싱
 - 사내 위키(Notion), 대화 로그(Slack), DB 등 흩어진 데이터 소스를 식별하기
- 인프라 구축
 - 실시간 데이터 연결을 위한 MCP 표준 적용하기
 - 효율적인 검색을 위한 Vector DB (RAG) 아키텍처 설계
- 데이터 정제
 - AI가 읽기 좋은 형태로 청킹 후, 메타 데이터 태깅

3. 아키텍처 설계 및 모델 선택

단일 모델을 쓰지, 여러 에이전트를 협업시킬지 결정

- **모델 티어링**
 - 비용 최적화를 위해 SLM(경량 모델)과 LLM(대형 모델)을 섞어서 쓰는 로직 설계하기
- **워크플로우 설계**
 - 에이전틱 루프(계획 → 실행 → 검토)의 단계를 정의하기
- **연결성 설정**
 - AI가 직접 실행해야 할 외부 도구(API) 리스트 확정

4. 반복적 개발 및 평가

가장 길고 중요한 단계; 코딩보다 프롬프트와 지능 맞추는게 집중

- **골든 데이터셋 구축**
 - 모범 답안지 50~100세트 만들기
- **프롬프트/파인튜닝 루프**
 - 프롬프트 수정하고 결과를 LLM-as-a-Judge로 자동 평가
- **지표 최적화**
 - 충실성(Faithfulness), 보안 준수(Security) 등의 점수가 목표치에 도달할 때까지 반복하기

5. 배포 및 거버넌스

실제 운영 환경에 AI를 올리는 단계로 보안이 핵심

- **보안 레이어 적용**
 - 개인정보(PII) 마스킹과 권한 기반 접근제어(RBAC) 활성화
- **가드레일 설정**
 - 부적절한 답변이나 탈옥(jailbreak)공격을 차단하는 레이어 활성화
- **Human-in-the-loop**
 - 최종 결과물을 인간이 컨펌할 수 있는 UI를 배포함

5. 배포 및 거버넌스

Human-in-the-loop

- 어떤 판단은 반드시 인간이 해야 하는가?
- 어떤 판단은 AI에게 위임해도 되는가?
- 인간은 승인자/ 감독자 / 개입자인가?

5. 배포 및 거버넌스

- 모든 자동화는 **실패**를 전제로 한다
- 모든 판단에는 **책임 주체**가 존재해야 한다
- 인간은 예외 처리자가 아니라 **구조적 구성원**이다

6. 모니터링 및 지속적 최적화

AI는 배포 후에도 성능이 변할 수 있음(Model Drift)

- **로그 분석**
 - 실제 사용자의 질문과 AI의 답변 로그를 수집함
- **비용 및 성능 감시**
 - 실시간 토큰 사용량과 지연 시간 (latency) 모니터링
- **지속적 학습**
 - 사용자의 피드백을 다시 골든 데이터셋에 추가하여 모델을 업그레이드

어떻게 자동화할 것인가? (X)
어디까지 자동화해도 괜찮나? (O)

6-stage AI Development Life Cycle (AIDLC)





**“우리는 곧 10인 이하의 1조짜리
회사들을 보게 될 것이다.”**

**“We’re going to see 10 person
\$1B companies pretty soon.”**

Sam Altman,
CEO of OpenAI

강의 듣느라 고생하셨습니다

강사 이준서(쑤스) | 이메일: hosohere@gmail.com