

SW 엔지니어로서 통련하기 위해 우리가 노력해야 하는 것들

Jungwoo Kim



Introduce the speaker

김정우

Jungwoo (Jayden) Kim

現) 안랩클라우드 서비스개발팀 리드

現) 그렙(프로그래머스) 데브코스 멘토

現) IT여성기업인협회 SW캠프 멘토

前) 모두투어 ERP개발팀 리드

웹, 서버, 클라이언트 앱을 개발하고 그 애플리케이션이 서비스 목표를 달성하도록 만드는 일을 하고 있습니다.

기술을 쉽게 풀어서 설명, 공유하는 것을 즐겨워 합니다.



Table of Contents

- 01** 좋은 SW 엔지니어란 무엇인가
- 02** 기술 역량의 지속적 성장
- 03** 커뮤니케이션과 협업
- 04** 효율적인 시간 관리와 생산성
- 05** 건강과 웰빙 관리

개발 역량

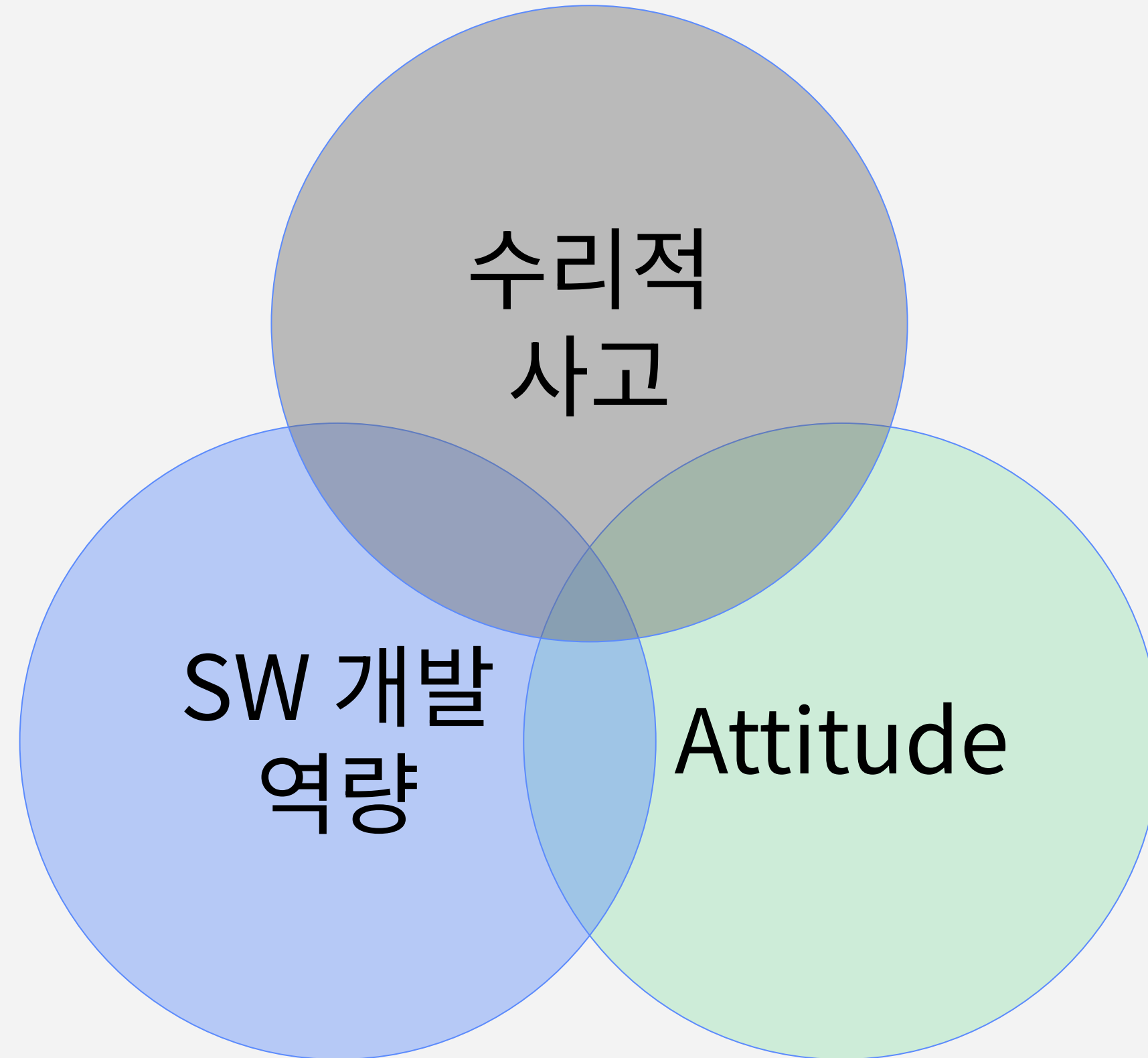
애틀튜드

Computational Thinking

01

좋은 SW 엔지니어란 무엇인가

좋은 SW 엔지니어에 대한 정의부터 내려보자



좋은 SW 엔지니어에 대한 정의부터 내려보자

🐱 문제 해결의 3요소



수리적 사고

수학적 사고
비판적 사고
데이터 분석적 사고
절차적 사고
문제 해결 능력

컴퓨터 공학,
소프트웨어 공학 지식
프로그래밍 언어 활용
능력 (패턴, 구조화)

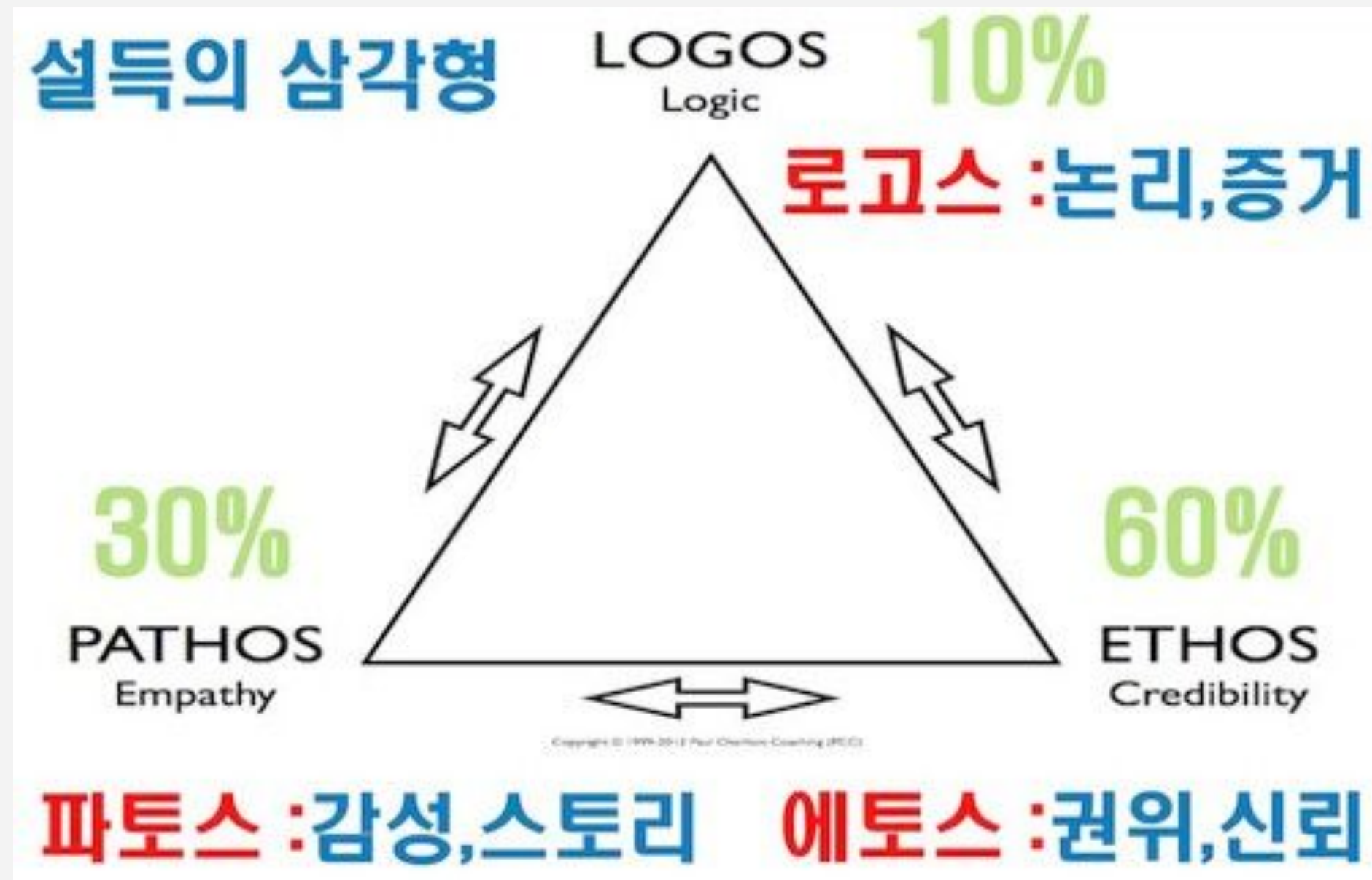
SW 개발
역량

Attitude

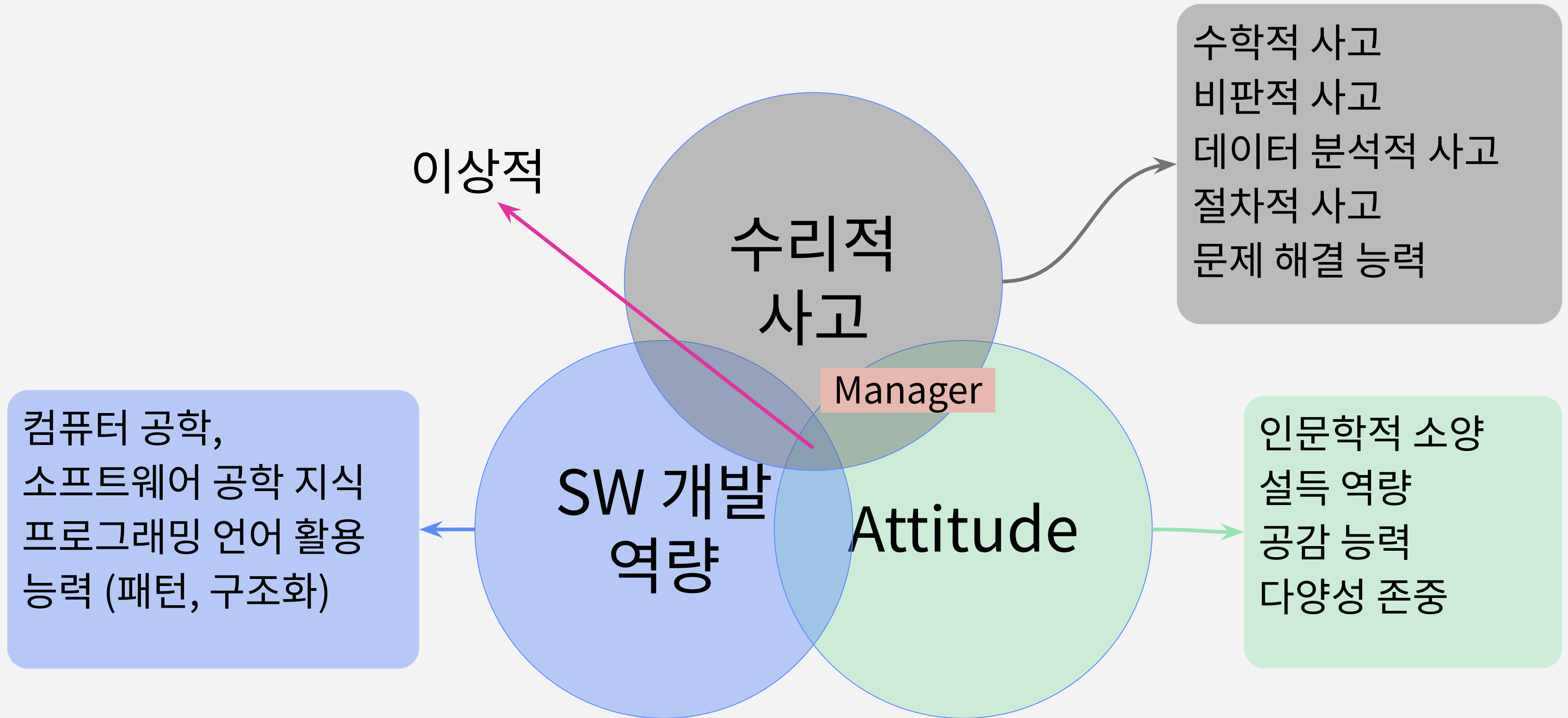
인문학적 소양
설득 역량
공감 능력
다양성 존중

좋은 SW 엔지니어에 대한 정의부터 내려보자

아리스토텔레스가 수사학에서 강조한 설득의 3요소



좋은 SW 엔지니어에 대한 정의부터 내려보자



클린 코드

애플리케이션 구현 패턴

시스템 아키텍처 패턴

02

기술 역량의 지속적 성장

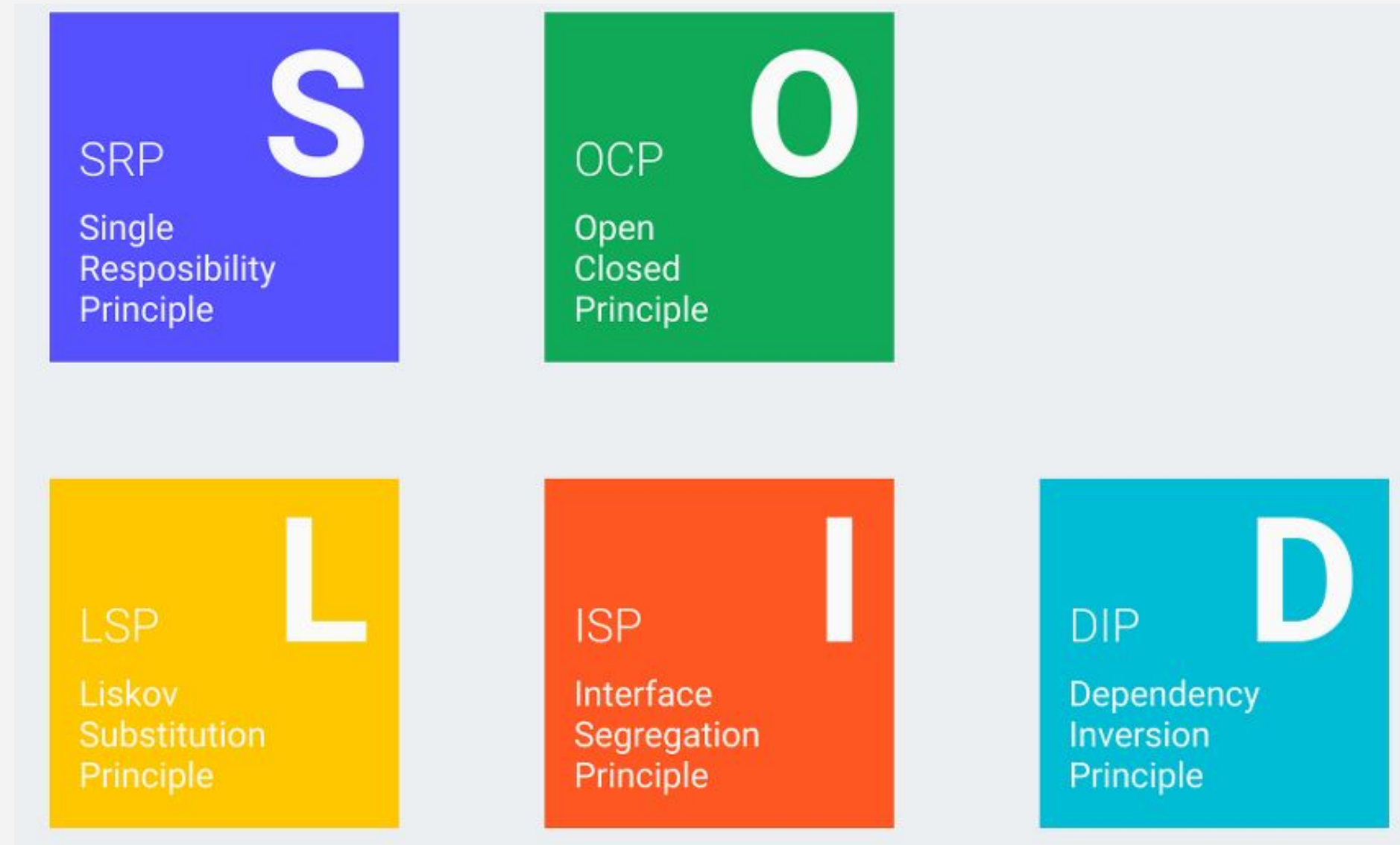
성장 가스라이팅 또 시작인가

클린 코드, 코드 품질에 대한 끊임 없는 탐색

- 좋은 코드는 무엇인가
 - > 코드를 처음 보는 사람도 동작을 직관적으로 파악할 수 있도록 작성된 코드
 - > 유지보수성 상승
- Coding Convention - 네이밍 룰, 들여쓰기, Lint, 주석 작성 방식 등
 - E.g., `MakeBillInvoice_RecentExpand_Save()`
 - > `SaveInvoiceRecent()`
- 함수의 인수는 적을수록 좋다, 작은 함수를 지향
- 중복 코드 생성 X, 조건문 간소화
- 하드코딩은 최소한으로

클린 코드, 코드 품질에 대한 끊임 없는 탐색

- 객체지향설계 5원칙 SOLID (단일책임, 개방폐쇄, 리스코프치환, 인터페이스분리, 의존역전) 을 실제 코드에 적용해보자.



디자인 패턴 실무 적용

- 작은 노력과 짧은 시간 이내에 애플리케이션의 성능을 향상시키고 유지보수성을 유지할 수 있도록 하기 위함.

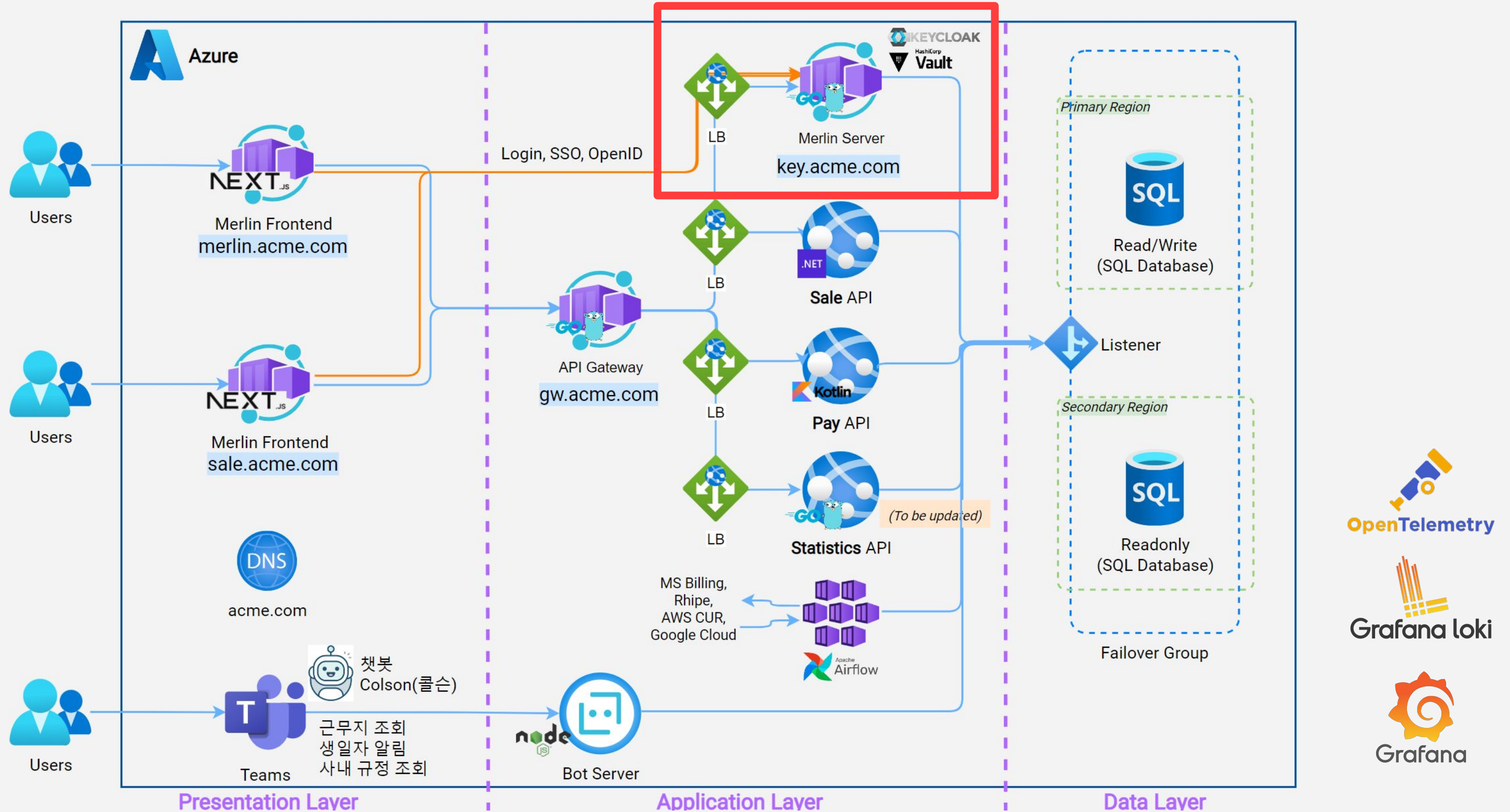
		Creational	Structural	Behavioral
Scope	Class	Factory Method	Adapter	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Proxy	Chain of Responsibility Command Iterator Mediator Memento Flyweight Observer State Strategy Visitor

출처: <https://circle.visual-paradigm.com/catalog/>

시스템 아키텍처 개선

- 전통적(?)이고 고질적인 장애 포인트를 개선하여 장애 대응 업무를 줄이고 생산성있는 업무에 집중할 수 있도록 하기 위함.
- 일단 단일장애지점(Sing Point of Failure)을 없애는 것부터 시작.
 - > 클러스터링 구성, 클라우드 PaaS 활용 등
- 장애가 쉽게 전이되지 않도록 해야 함.
 - > 서킷 브레이커 패턴
- CQRS 패턴
- HA (고가용성) 솔루션과 DR (장애복구)에 대한 시나리오가 필요함.

시스템 아키텍처 개선



시스템 아키텍처 개선

Keycloak Clustering + Azure Application Gateway

개요

키클락의 단일 장애 지점을 없애기 위하여 키클락 클러스터링 구성에 대한 연구를 진행하였다.

그 결과 키클락 2대의 클러스터링 작업 완료 및 Azure Application Gateway를 연동하는 것에 성공하였으며 그 방법과 이 구성의 이유를 설명하고자 한다.

키클락 버전은 27 버전으로 진행하였다.

구성 - 키클락

키클락 클러스터링을 구성하기 위하여 Azure VM을 2대 생성하였다.

IP는 다음과 같다. 나는 내부IP를 이용하여 클러스터링 했다.

VM1, Keycloak1: 10.0.0.5

VM2, Keycloak2: 10.0.0.6

또한 포트 구성은 다음과 같다.

API 포트 : 8080

Infinispan Bind Port : 7800

Infinispan Failure Check Port : 57800

Failure Check Port 는 따로 지정하지 않으면 Infinispan Bind Port + 50000 이 되므로 그냥 두어도 된다.

다만 외부 IP를 이용하여 연동할 경우 포트포워딩은 해주어야 한다.

|just start typing...

Infinispan 컨피그를 수정해야 한다.

하라는 대로 하지 않는다.

- 하라는 대로 하기 전에 이것을 왜 해야 하지? 어떤 임팩트를 가져올 수 있지? 고민을 해야 한다.

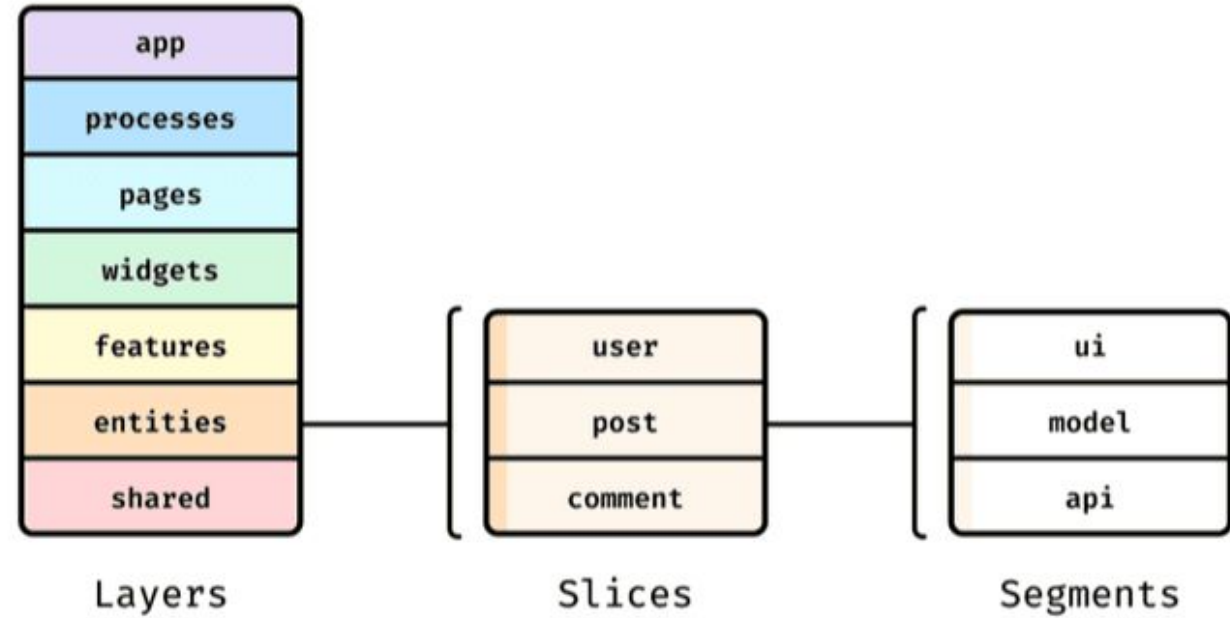
- 시켜서 하는 일 외에도, 남들이 생각하지 못한 것을 대비할 수 있는 부분까지 미리 처리하기.

(그렇게 그게 내 일이 되고...)

새로운 기술 짭먹하기

- 이미 수없이 많은 언어와 프레임워크, 라이브러리가 있지만 새로운 것이 계속 나오고 있음.
- 프론트엔드의 기술 트렌드도 빠른 편, 특히 AI의 속도는 엄청 남.
- 새로운 서비스, 새로운 라이브러리, 새로운 DB 가 나오면 직접 테스트해보고 장단점 기록해서 개인 GitHub에 올려두자
(다만, 트렌드를 맹신해선 안됨)

FSD 아키텍처



FSD 구조 정의

- 레이어 계층
- shared → entities → features → widgets → pages → app

app

- Providers
- 실제 nextjs app을 그대로 옮겨도됨

pages (views)

- 라우트 페이지

widgets

- 도메인단위로 여러페이지에서 사용될 수 있는 컴포넌트
- 예) 조직 그리드 위젯은 여러 페이지에서 사용될 여지가 있다.

SD팀 / NextAuth.js

Normal text | B I ... | A | | | | | | | | | | | | | |

API Route

- app/api/auth/[...nextauth] 파일을 생성해야 한다
 - NextAuth.js의 dynamic route handler 및 모든 global NextAuth.js configurations을 포함한다
 - /api/auth/* 에 대한 모든 request는 (signIn, callback, signOut, etc.) NextAuth.js에 의해 자동으로 처리된다

확장성

확장성

- NextAuth.js의 callbacks을 통해 인증 흐름의 다양한 부분에 연결할 수 있다
- ex) 로그인에서 프론트엔드 클라이언트 측으로 값을 전달하려면 다음과 같이 session 및 jwt 콜백을 조합하여 사용할 수 있다

```
1 ...
2 callbacks: {
3   async jwt({ token, account }) {
4     // Persist the OAuth access_token to the token right after signin
5     if (account) {
6       token.accessToken = account.access_token
7     }
8     return token
9   },
10  async session({ session, token, user }) {
11    // Send properties to the client, like an access_token from a provider.
12    session.accessToken = token.accessToken
13    return session
14  }
15 }
16 ...
```

Options

- session 객체
 - 종류
 - jwt session (JSON Web Token)
 - 모든 세션 정보를 암호화된 JWT로 쿠키에 저장

코드 리뷰

기술 지상주의 경계

일정 공유

명확한 말 표현 습관

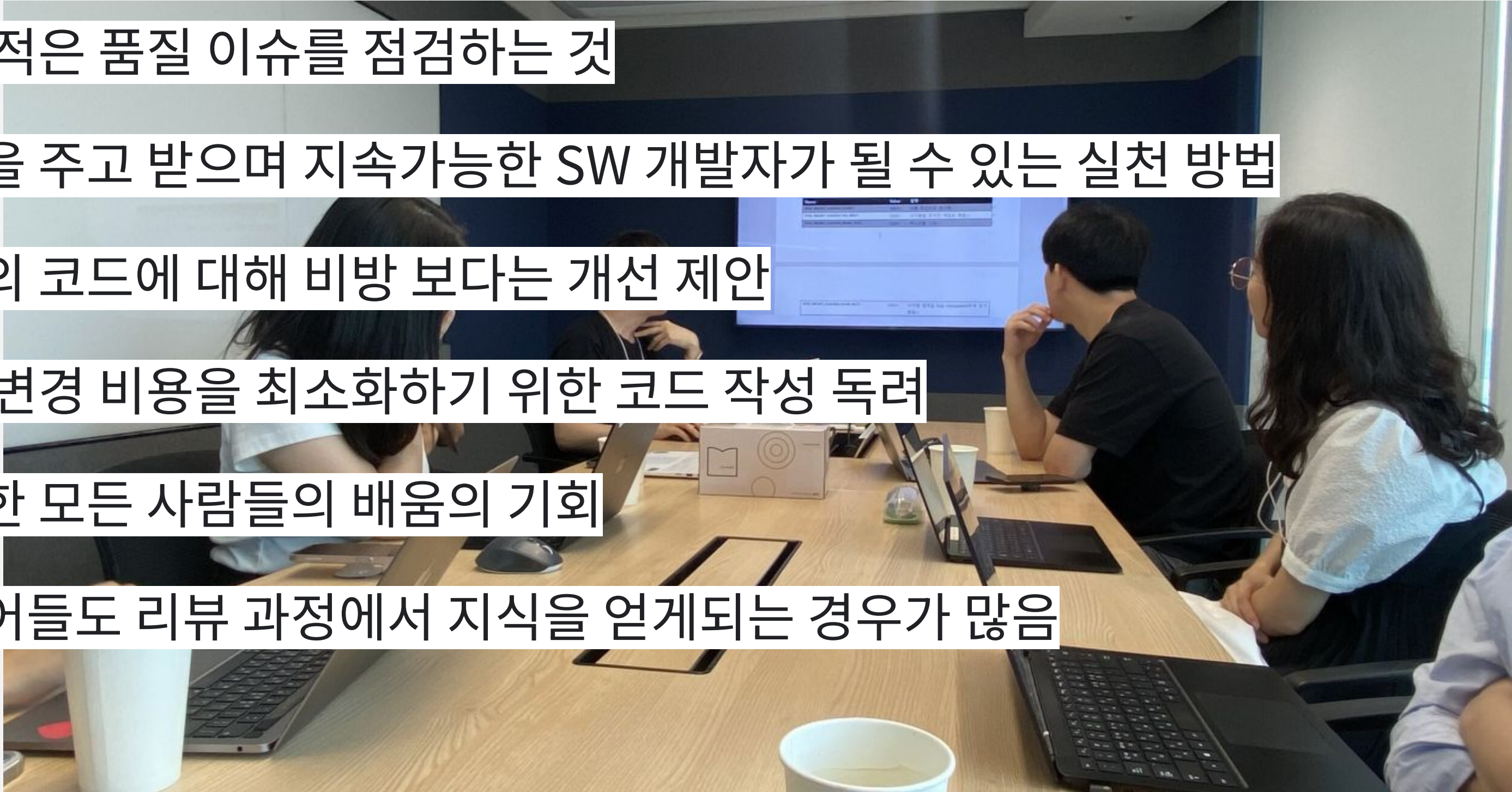
03

커뮤니케이션과 협업

우린 모두 좋은 동료와 일하고 싶다.

동료와의 효과적인 코드 리뷰 문화

1. 주 목적은 품질 이슈를 점검하는 것
2. 배움을 주고 받으며 지속가능한 SW 개발자가 될 수 있는 실천 방법
3. 타인의 코드에 대해 비방 보다는 개선 제안
4. 향후 변경 비용을 최소화하기 위한 코드 작성 독려
5. 참여한 모든 사람들의 배움의 기회
6. 리뷰어들도 리뷰 과정에서 지식을 얻게되는 경우가 많음



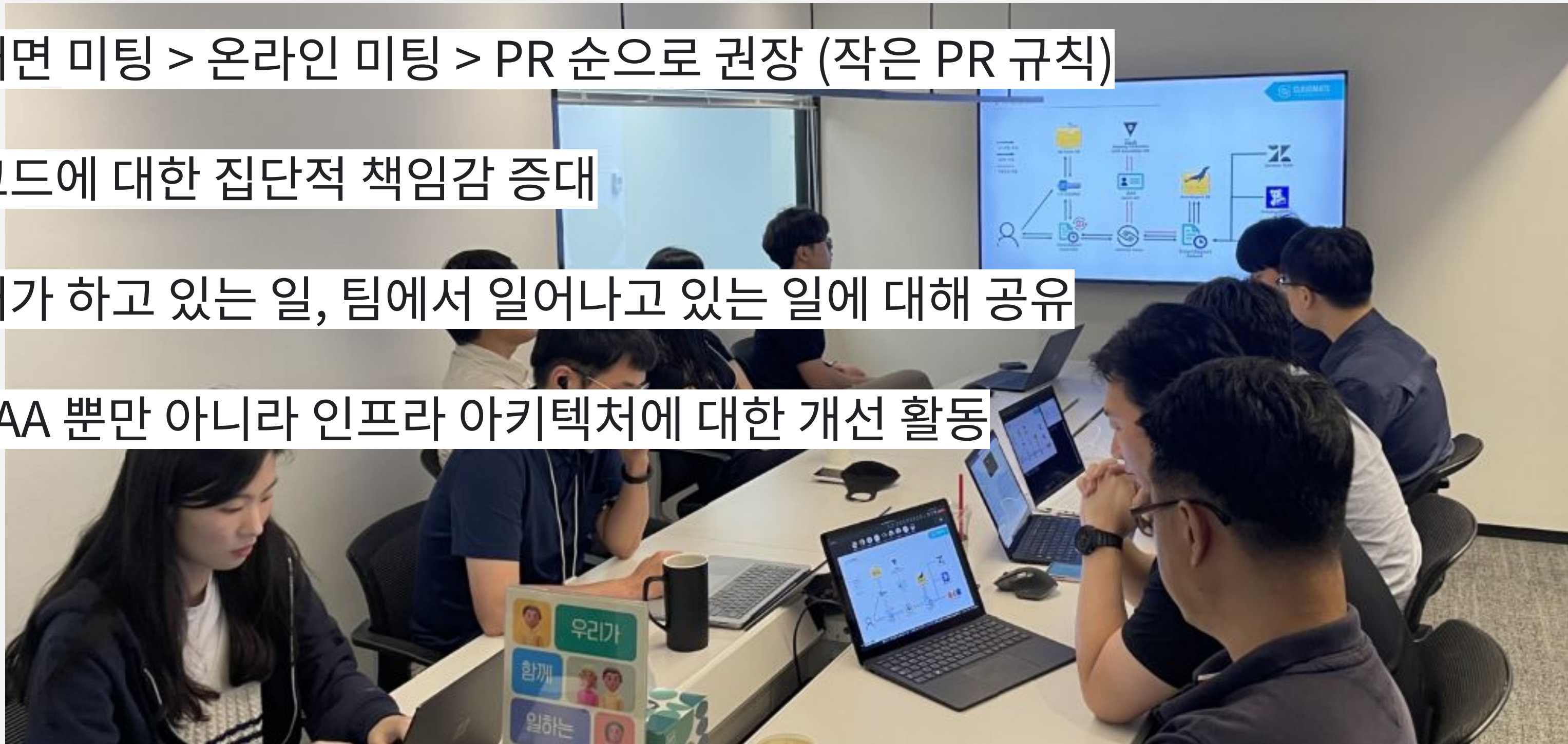
동료와의 효과적인 코드 리뷰 문화

7. 대면 미팅 > 온라인 미팅 > PR 순으로 권장 (작은 PR 규칙)

8. 코드에 대한 집단적 책임감 증대

9. 내가 하고 있는 일, 팀에서 일어나고 있는 일에 대해 공유

10. AA 뿐만 아니라 인프라 아키텍처에 대한 개선 활동



기술로만 해결하려고 하지 않아야 한다

- Goal을 위해서 활용 가능한 모든 방법을 동원해야 함.
- 기술로만 해결하려고 하면 오히려 더 시간이 오래 걸리는 경우도 많음.
- 기술 우월주의 경계.

E.g., 영업이 뭘 알아. 운영팀이 뭘 알아. 대표님이 뭘 알아.

기술로만 해결하려고 하지 않아야 한다

- 많은 개발조직에서 확장하는 것에만 몰두하고
유저가 좋아하는 것에 집중하지 않는다.

Instagram co-Founder Kevin Systrom



"결국 중요한 것은 제품 시장 적합성이며,
그것이 가장 중요합니다."

"유저들이 제품을 좋아하기만 한다면요."

"5만명이 언제 당신의 제품을 사용할지는
나중에 고민해도 됩니다."

"왜냐면 그때쯤이면 그 문제를
갖고있다는 자체에 행복할테니까요."

출처: <https://youtu.be/V27XkmVPqYQ>

협업 문화 만들기

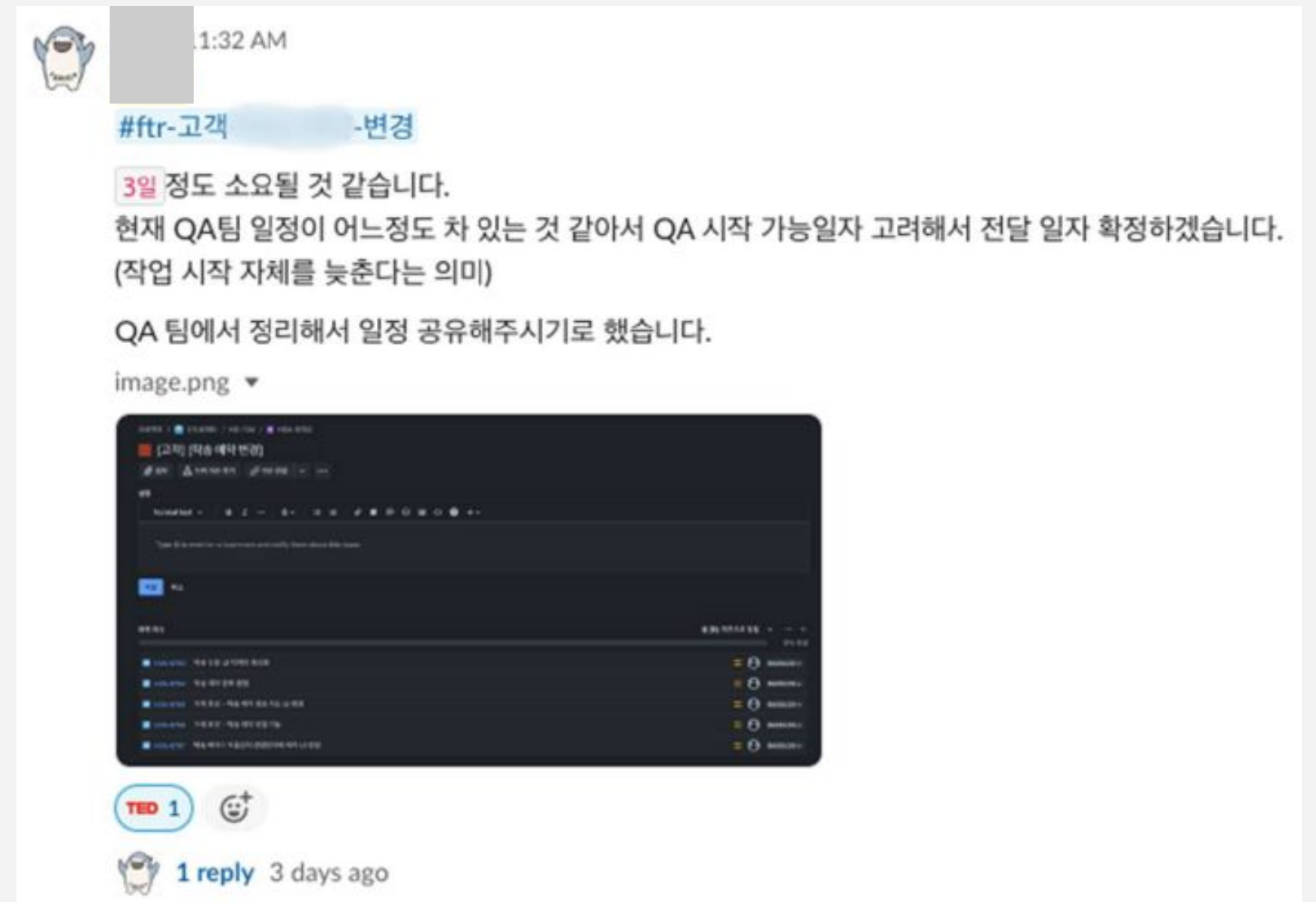
1. Agile 개발방법론 (빠른 Waterfall 아님)
2. 하나의 feature를 작게 구현하고 점진적으로 업그레이드
3. Agile process를 **조직 성과 관리 측면**으로만 바라보면 구성원들이 **피폐**해짐
4. 실패했더라도 과정 속에서 배운 것들 회고하고 보완 - **좋은 실패** 🤔
5. 각 프로젝트에 맞는 sprint iteration 설정 (2weeks / 4weeks)
6. 주기적인 Scrum meeting을 짧게 진행 (standup meeting)
7. 본인 R&R이 기본이고 내 주변 업무도 내 업무 처럼

일정 공유

- **예측 가능**하게 만들어주기
- 지금 하고 있는 작업이 언제쯤 끝날 것 같은지 공유
- 리더/기획자로 하여금 일정을 산정할 수 있게 만들어준다
- 오래 소요되는 작업이라면 **중간공유**를 한번 한다
- 일정에 문제가 생길 것 같으면 바로 공유
- 일정을 고민하고 일을 하다보면 이런 사람들이 결국 데드라인을 잘 지킨다

일감을 구상할 때 목표, 기간, 결과물, 공유자를 생각한다

- 해당 일에 대한 목표, 목적, 예상 기간, 예상 결과물, 공유대상자를 생각하고 일을 시작한다.
- Todo를 최신화하고 reminder 걸어놓는다.
- 이거 내가 왜 시작했지? 다른 일감에 따라 좌지우지되거나 흐지부지됨.

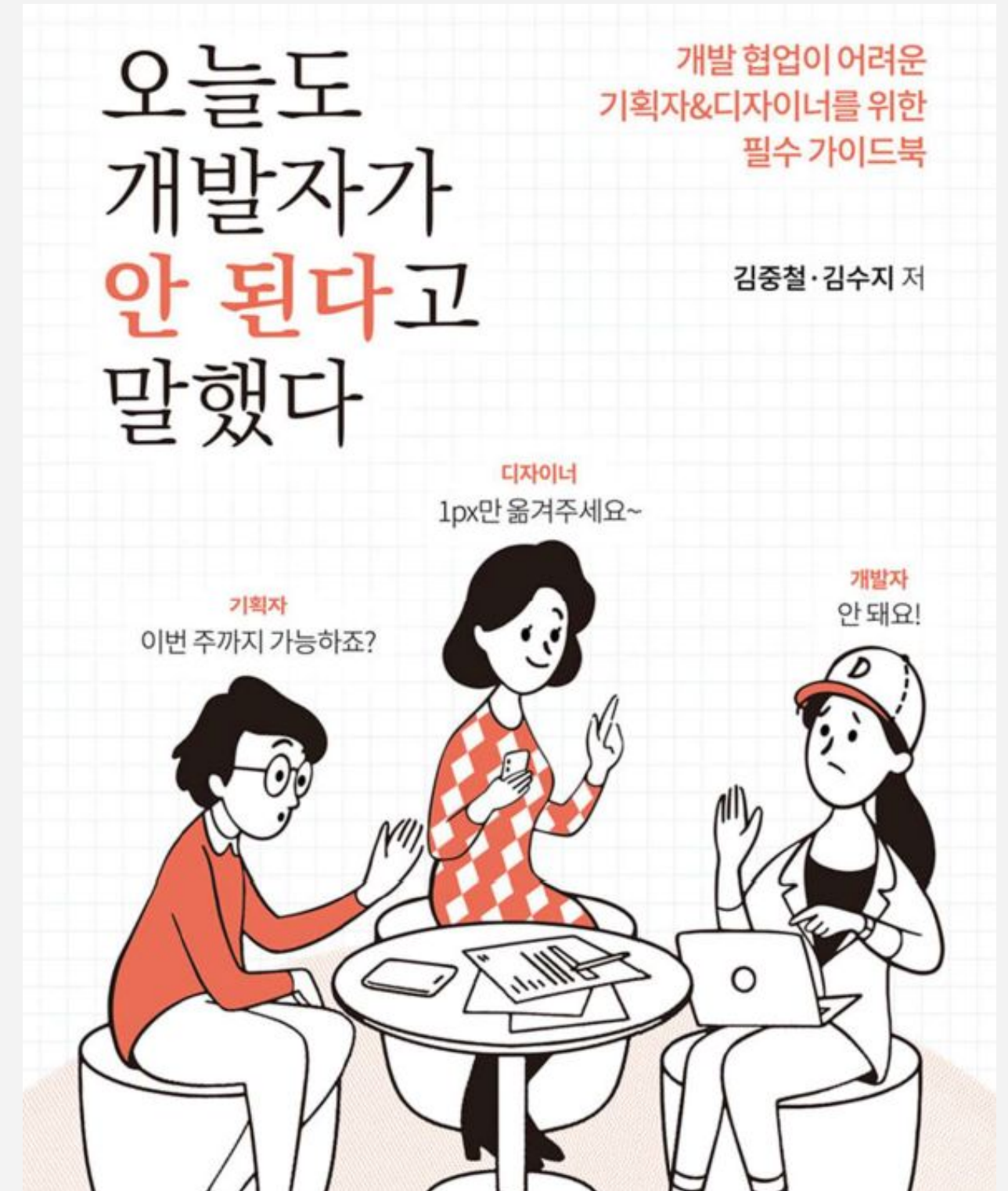


좋은 인간이 되자

- 인문학적 소양이 필요함
- > 인간과 사회를 이해하고 성찰하는 능력
- > 핵심은 공감 능력
- > 그럴 수도 있지. 그때는 그럴만한 이유가 있었을꺼야.
- 업무는 수직적으로 할 수 있지만 인간적인 관계는 수평적으로 가는 것이 좋다
- 좋은 선임, 동료, 후임이 되자
- 기분이 태도가 되어선 안된다
- 남 뒷얘기하는 사람, 거짓말하는 사람, 매사 부정적인 사람은 경계하자

非IT 직원과 원활하게 소통하려면?

- “안된다”고 하지 않기
- 선택할 수 있는 대안을 여러개 제시하고 각 대안들에 대한 pros and cons을 명확히 정리해서 공유한다
(타인의 눈높이, 관점에서)



명확한 단어와 문장 사용


- 말 하는 방식 매우 매우 중요

<p>로 승리하기</p> <p>일, 내 일 선긋지 않습니다. 이는 같은 목표를 추구하는 필이기 때문입니다.</p>	<p>매너있게 분명하게</p> <p>마주한 상대의 감정을 높은 수준으로 배려하면서, 의견을 명확하게 전달합니다. 한번에 알아들을 수 있도록 배경과 맥락을 충분히 공유합니다.</p>	<p>규율있는 자율</p> <p>규칙이 없어도 회사에 이익되는 행 동행합니다. 동료에게 신뢰를 주 사람이 됩니다.</p>
--	---	--

인정할 것은 인정하자

- 인정할 줄 알아야 한다.
- 문제를 널리 알린다.

tech.inflab.com 채용 인프런 랠릿



2022년 1월 100% 할인 이벤트 장애 부검

posted by [향로\(HyangLo\)](#), February 01, 2022
14 min read

backend

Postmortem Service

안녕하세요 인프런의 향로입니다.
인프런 서비스는 2022년 1월 3일 ~ 17일까지 지속적으로 서비스를 사용할 수 없는, 역대급 장애가 발생했습니다.

이번 장애 기간동안 정말 많은 분들의 질책을 들으면서 반성을 했습니다.
특히 퇴근 후 1~2시간 공부밖에 못하시는 분들께서는 그날의 학습을 모두 포기하게 되는 경험을 드리게 되어서 더욱 복잡한 마음이 들었습니다.

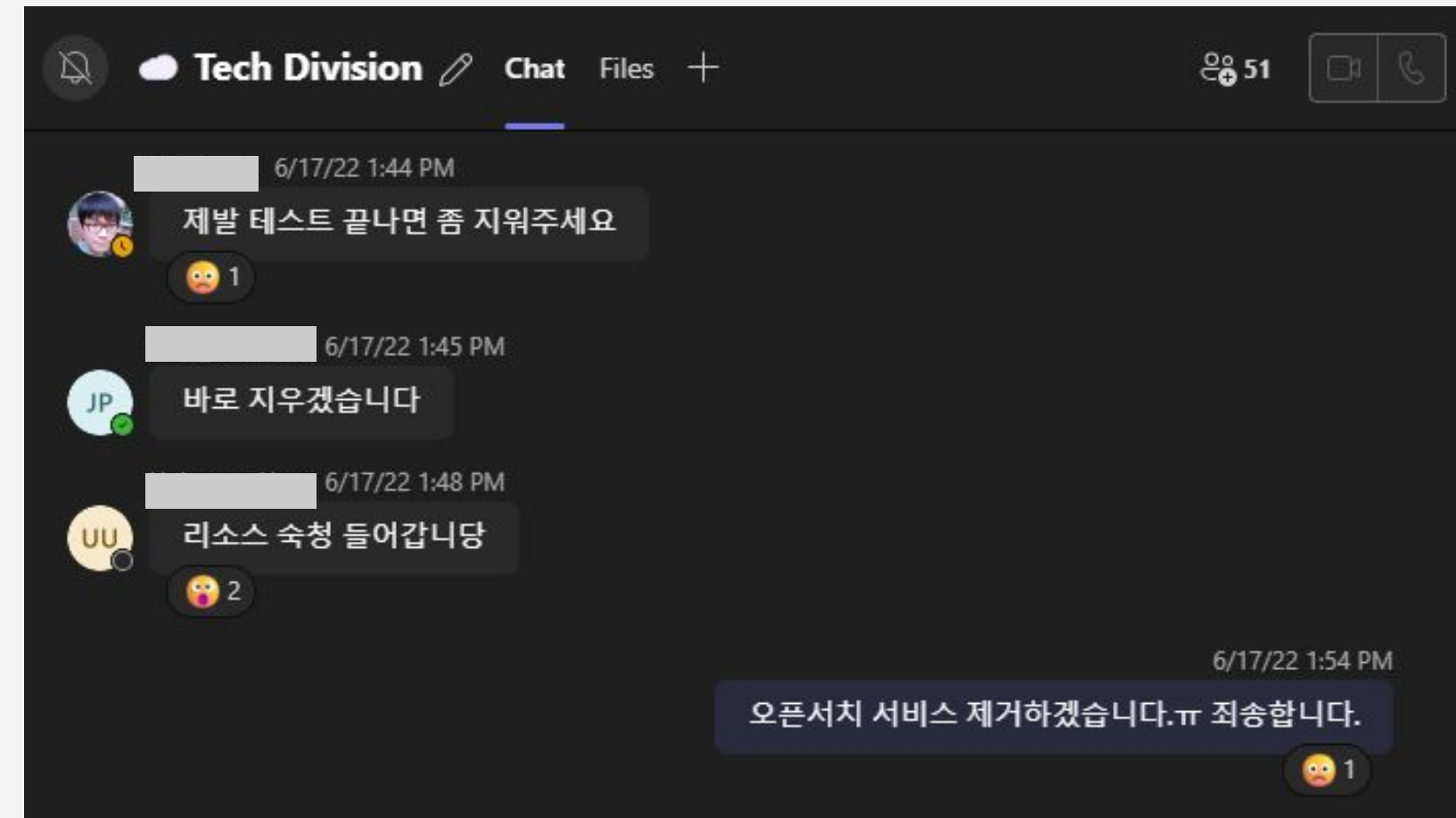
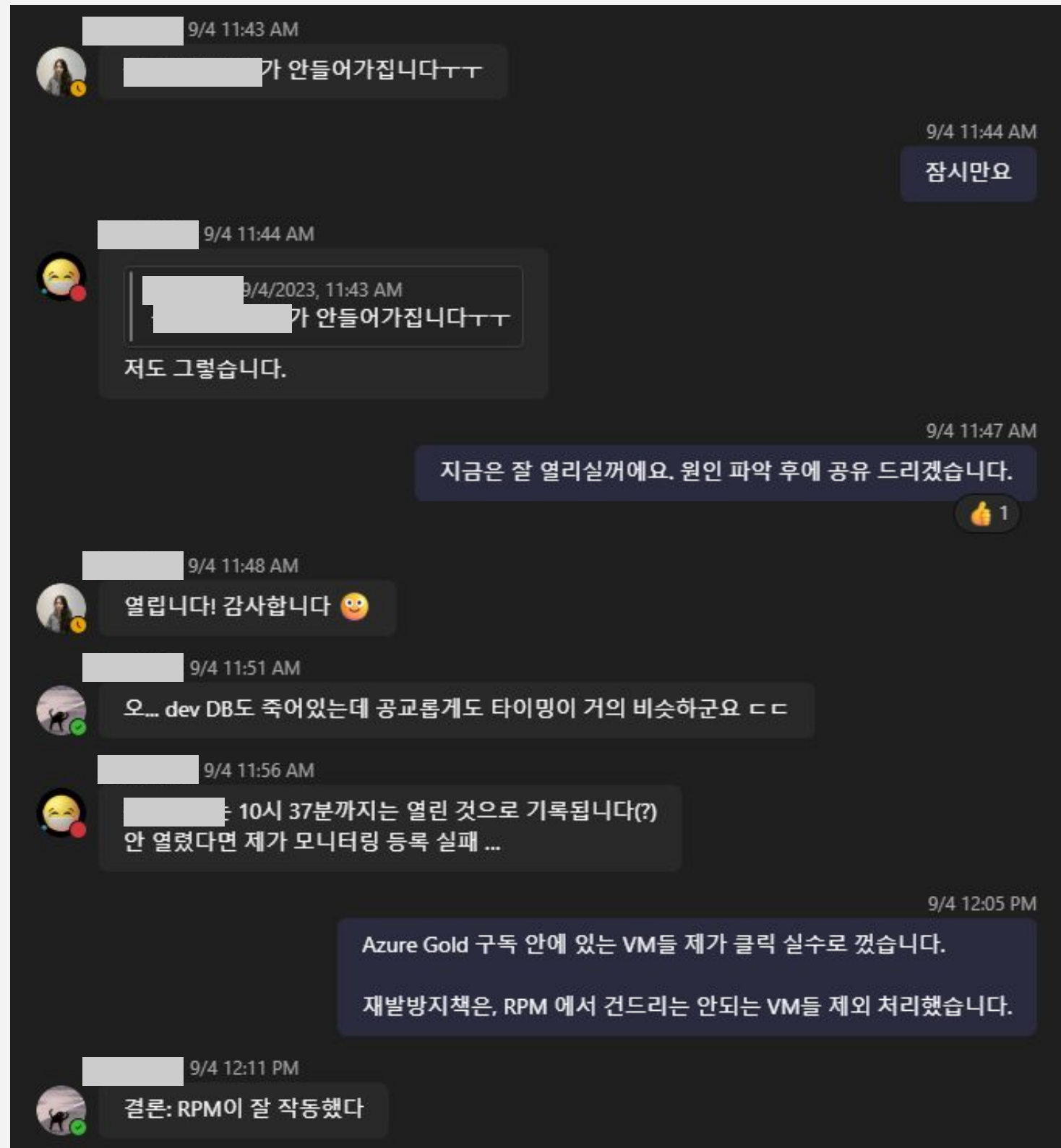
이번 포스팅을 통해 왜 이렇게 오랜 기간 서버 장애가 발생했는지, 근본 원인이 해결 된 것은 맞는지 등의 의문을 해소하려고 합니다.

이런 장애 부검은 앞으로도 지속적으로 유지하면서 인프런 서비스가 개선되고 있음을 투명하게 공유드리도록 하겠습니다.

다시 한번 서비스 사용에 불편을 드려 정말 죄송합니다.

- 1. 발단
- 2. 장애 회고
 - 2-1. 서버 구조
 - 2-2. 1월 3일 장애
 - 2-3. 1월 4일 장애 수정 지연
 - 2-4. 1월 7일 장애
 - 2-5. 1월 17일 장애 (최종)
 - 2-6. 근본 원인 분석
- 3. 마무리

인정할 것은 인정하자



내가 속한 회사 조직 이해하기

- 회사의 미션과 비전, 내가 속한 팀의 미션과 비전은 무엇인가
- 조직 문화, 조직도
- 수평 조직 이해하기



문서화의 중요성

어차피 혼자서는 다 할 수 없음

회고

04

효율적인 시간 관리와 생산성

당신은 혼자가 아니다.

혼자서 다 하려고 하지 않기

- 일이 잘 풀리지 않을 때 혼자 전전긍긍하지 않는다
그러다 데드라인 다 되서 터트리는 사람도 많음
- 적극적으로 주변 사람들에게 도움을 요청
- 모르는 것은 모른다고 하고, 부탁할 것은 부탁을 하자
- 좋은 질문을 해야 좋은 답을 얻을 수 있음. (상대방의 관점, 눈높이 등)
- 누군가 나에게 질문하면 성심성의껏 답 해준다 (개인의 성장 -> 조직의 성장)
- 집단지성의 힘을 믿자
- 의외로 어떤 사람이 짠- 하고 나타나서 쉽게 잘 해결해주는 경우도 꽤 많음
- Cherry Picker는 되지 말기

암묵지 잘 기록하기

- “기록은 기억을 지배한다” / “적자생존”:

적는 자가 생존한다.

- 회사에서 업무를 하면서 발생하는 암묵적 지식(Tacit knowledge)을 어딘가에 잘 기록해놓자 (위키 등)

E.g., 애플리케이션 아키텍처, 업무 처리 매뉴얼, 서비스 흐름도, 각종 피쳐의 다양한 논의 내용

[FE] 2025.02.12 billing3.0 회의록

2.12 빌링 회의 아젠다

QA

- 데이터그리드 페이징네이션 안됨
 - 쌍화살표 제거
 - 화살표 넘어가면 그 다음 첫번째 페이지 표시하기
 - 넘어갈 페이지가 없으면 아이콘 안보여주기
- 파트너사 계정 받아서 테스트 진행
 - Account 타입 확인 해보기
 - 실무자 QA 내용 확인해보기(팀장님께 물어보기) - [▲ Log in with Atlassian account](#)
- 사용량 조회 페이지에서 일단위 조회 에러가 안날 수 있도록 ui로 막기
 - 하은님 작업 중
- 에러 표시할 수 있는 공통함수 만들(하은) -> 설명하고 다같이 해당함수 사용하기
 - react-query를 이용해서 에러처리할 수 없는지도 탐색

```
1 // 해당 부분의 중복도 제거하고 싶음
2 if (isError && error?.message) {
3   if (prevErrorRef.current !== error.message) {
4     const formattedMessage = getErrorMessage(error.message)
5     snackbar.pushSnackbar(formattedMessage, { variant: 'error' }) // 스낵바 출력
6     prevErrorRef.current = error.message // 현재 에러 저장
7   }
8 }
9
```

git branch 전략

- task 완료 후 dev에 머지하는 전략이면 현재 배포날에는 총 n번이 배포됨 -> 이 문제에 대해 어떻게 전략을 가져갈 지 논의필요.
- **master**: 기준이 되는 브랜치로 제품을 배포하는 브랜치
- **develop**: 개발 브랜치로 개발자들이 이 브랜치를 기준으로 각자 작업한 기능들을 Merge
- **feature**: 단위 기능을 개발하는 브랜치로 기능 개발이 완료되면 develop 브랜치에 Merge
- **release**: 배포를 위해 master 브랜치로 보내기 전에 먼저 QA(품질검사)를 하기위한 브랜치

암묵지 잘 기록하기

- 모두가 고민했지만 누구도 명확하게 정의내리지 못했던 것들에 대해서 정의
- 규칙이 없는 것에 규칙을 만들고, 이름이 없는 것에 이름을 만든다
- 누군가에게 큰 도움이 된다 -> 나중에 나의 시간도 절약할 수 있게 된다

2025.1.21 고객사, 파트너사, ACME의 User Flow 요약

공통 사항

- 사이드바:
 - 권한 상관없이 항상 표시.
- 글로벌 트리 선택:
 - 고객사: 자기 회사만 표시됨, "전체 선택/해제" ui는 보이거나 실질적으로 불필요.
 - 파트너사/ACME: 전체 트리 표시.
- 대시보드:
 - + :: • 검색: 모든 권한에서 가능.
 - 최근 3개월 청구액 카드: 표시.
 - 당월 청구 상세 카드: 표시.
 - 최근 12개월 청구액 카드:
 - 고객사: 자기 회사 데이터만 표시됨 (단일 스택 바).
 - 파트너사/ACME: 모든 데이터 표시.
 - 사용량 조회:
 - 검색: 모두 가능.
 - 사용량 조회 그래프: 표시.
 - 드릴다운(추후 고려): 상세 사용량 표시 가능.

고객사

- 청구서 관리:
 - 검색: 가능.
 - 진호 선택/상태 저장 버튼/청구서 일괄 생성 버튼: 표시 안 됨.
- 그리드:
 - 한 달에 한 건의 데이터만 표시.
 - 상태: 항상 "확정".
- 액션 버튼:
 - 첫 번째 그리드: 상세 내용.
 - 두 번째 그리드: NCP 리소스 상세(참고용, 가격이 다를 수 있음).

회고

- 회고(Retrospection)는 마치 나만의 **오답노트**와 같다.
- 다음에 실수하지 않으려면 오답노트를 작성하고 발전해야 합니다.
- 프로젝트에 참여한 인원들과 함께 돌이켜보는 회고시간을 가져야 배움이 남고 성장할 수 있습니다.

회고

🔍 검색

🕒 업데이트 65

📁 모든 팀스페이스

⚙️ 설정과 멤버

➕ 새 페이지

📁 안드로이드팀

- 📄 온보딩
- 📄 How to manage
- 📄 How to use
- 📄 Roadmap
- 📄 Snippet
- 📄 Android Weekly Opi...

🕒 회고 기록

- 📄 회고
- 🕒 2023.08 팀회고
- 🕒 2023.07 팀회고
- 🕒 **2023.06 팀회고**
- 🕒 2023.05 팀회고
- 🕒 2023.04 팀회고
- 🕒 2023.03 팀회고
- 🕒 2023.02 팀회고

🕒 2023.06 팀회고

📄 Board view +

Keep 6

- 앨런이랑 짝들하게 같이 개발할 수 있어서 즐거웠음
👤 laco
💬 1
- 도메인 로직을 도메인 모델에 작성하자
👤 laco
- 상태 변경 소스로서의 원샷 및 스트림 API
👤 laco
- Compose 시작
👤 olaf
👤 olaf
- 어느 순간 테스트 코드를 짜는게 자연스러워진

Problem 9

- 피쳐일정 회고
👤 laco
- [달러] ██████████ 변경 되었을 때 일괄 반영이 안되어서 작업을 5번 했다
👤 laco
- MarkXXX 갯수가 심상치 않다
👤 laco
- 번호판 입력 버그로 많은 고객분께서 오류를 경험하고 지표 하락으로 이어졌을까?
👤 laco
- Phone Number... 🤔
👤 laco

Try 7

- 도메인 로직 관련해서 본인이 생각하는 best practice를 PR로 만들어서 공유
👤 olaf
- NeedShow & Mark&&XXX
👤 laco
- 컴포즈 공식문서 보기
👤 laco
- Compose 학습주간
👤 laco
- Flow<DataResource<T>> 대신 T 를 반환하는 UseCase 추가
👤 olaf 👤 laco
👤 laco

회고



AhnLab Cloud
가즈아!!!!



존경하는 대주주님



Liked - 만족스러웠던 점	Learned - 깨달았거나 배웠던 점	Lacked - 놓쳤거나 부족했던 점
<p>차음 장비는 기술 스펙으로 고객이 원하는 기능을 구현하였다. SAM (segment-anything-model) 라이브러리, fabric.js, vue 등..</p> <p>기존에 하지않은 SM 임무를 시작부터 인수인계를 받고 현재까지 문제없이 잘 진행 된 점</p> <p>회사가 안랩에게 인수</p>	<p>검증된 오픈소스더라도 시명적인 버그가 존재할 수 있다는 것을 배움 (fabric.js의 브러시) - 너무 라이브러리에 의존하지말자</p> <p>무차별적인 프레임워크 도입은 오히려 독이되는 듯 (lge-pri 프로젝트들)</p> <p>개발환경의 중요성</p> <p>Wasm 최고</p>	<p>Git branch 전략</p> <p>멘탈 관리</p> <p>이해하지 못하고 지나간 기술</p> <p>billing3.0 프로젝트를 준비를 하지 못함</p> <p>canvas 에 대한 지식 부족, vue 생명주기, 렌더링과 같은 기본적인 지식 부족으로 인한 이슈</p> <p>상대적으로 기술적인 성장이 없어 아쉬움</p>



건강한 자세와 운동

스트레스 관리

05

건강과 웰빙 관리

일과 삶의 균형 유지하기

SW 엔지니어를 위한 건강한 자세와 운동

- 건강하지 않으면 아무 것도 할 수 없습니다.
- 좋은 책상, 좋은 의자, 인체공학적 마우스
- > 200만원짜리 H사 의자에 앉아도, 앉아있는 자세가 사우나 자세면 말짱 광
- 모니터와 키보드, 마우스의 최적의 위치를 찾으세요
- 허리는 의식적으로 자주 세워주세요
- 아침이나 업무 틈틈히 허리, 어깨, 목, 손목, 무릎을 풀어주세요

스트레스 관리와 마인드 컨트롤

- 우리는 돈을 받고 일해주는 프로이기 때문에 일은 원래 힘든 것.

이것 또한 지나가리라.

답답함, 욕구 불만, 번아웃 증후군

- 일이 힘든 것 보다 사람이 미우면 훨씬 힘들어짐.

일과 삶의 균형 유지

햇볕이 드는 곳에서 산책, 활동

카타르시스가 느껴지는 좋은 공연, 영화, 스포츠 경기 직관

고전 - 맥베스, 햄릿, 오만과 편견, 죄와 벌, 구운몽, 군주론

사내 또는 사외 멘토 등 다양한 사람들과 커피챗

Thank You

