



0. 프로젝트에 들어가기 앞서

[프로젝트란 무엇일까?](#)

[협업이란?](#)

[프로젝트를 시작하면 제일 먼저 해야 할 것](#)

[서비스 구조를 그려보기](#)

프로젝트란 무엇일까?

| 정해진 기간 안에, 여러 사람이 함께, 하나의 결과물을 완성시키는 과정

여기서 중요한 것은 3가지라고 생각합니다.

- 정해진 기간
- 여러 사람
- 완성

팀 프로젝트는 뭐가 다를까?

토이 프로젝트. 즉, 혼자서 하는 프로젝트는 혼자서만 진행하기에

스파게티 코드여도, 문서가 없어도 내 스스로 알고 있으면 상관 없습니다.

하지만, 팀 프로젝트의 경우 여러 명이 작업하기에 나 뿐만 아니라 남이 봐도 이해할 수 있어야 하고

결정에 대해 설명이 필요합니다.

그래서 협업하는 방법과 프로젝트를 진행하면서 반드시 해야 할 일, 문서화는 왜 필요한지 등에 대해 설명해드리도록 하겠습니다

협업이란?

협업을 오해하는 경우가 많습니다.

각자 파트에서 맡은 기능 열심히 개발 해서 완성해가는 거 아닙니다.

이건 동시 작업이지, 협업이 아닙니다 ㅠ

협업은 '같은 결정을 같은 이유로 이해할 수 있는 것'입니다.

파트를 코드를 나누는게 아니라

- 왜 이 기능을 만드는지
- 왜 이 구조를 선택했는지
- 왜 지금 이 순서인지

등 모두가 알고 있어야 합니다.

이렇게 알게 되고 있더라도 협업이 깨지는 순간들이 옵니다.

머릿속으로만 계획에 있고 말로만 합의 했을 때,

기준이 사람마다 다른 경우,

의견만 있고 결정권자가 없을 경우.

그렇기에 협업에서는 **기준**이 필요합니다.

- 기능 완료의 기준
- 코드 리뷰의 기준
- 일정에 대한 기준
- 의사결정에 대한 기준

그리고 그 기준을 세웠더라면 **기록**을 해야 합니다.

저희가 만약 얘기 한 것을 잘 기억하면 정말 좋겠지만,

저의 경우 나타하기도 하고 결정을 까먹는 경우가 많습니다.

- 회의 내용
- 결정 사항
- 변경 이유

말은 사라지지만 기록은 남아있으니 꼭 기록하는 걸로..!

마지막으로 정리할 겸 좋은 팀은

- 결정은 반드시 기록한다
 - 기준이 없는 작업은 시작하지 않는다
 - 모르면 묻는 게 아니라 공유한다
 - 문제는 개인이 아니라 팀의 문제로 본다
- ⇒ 잘하는 사람이 많은 팀이 아니라 같은 방향을 보는 팀이 잘 된다.

프로젝트를 시작하면 제일 먼저 해야 할 것

프로젝트를 시작하게 되면 우선 레포부터 파고 베이스 코드부터 쳐야 할 것 같지만 해커톤이 아닌 이상 저는 개인적으로 목표부터 정하는 게 가장 중요하다고 생각합니다.

왜냐하면

1. 목표가 없는 구현이 된다
2. 사람마다 생각하는 완성 기준이 다르다
3. 기획은 되돌리기 어렵다

그렇기에 시작한다면 다음과 같이 진행하면 좋다고 생각합니다

1. 프로젝트의 목표를 명확하게 정한다 → 짧으면 더 좋고
2. 성공 기준 즉, 끝을 정한다
3. 기간과 마일스톤을 정한다
4. 역할과 책임을 명확히 한다 → 주 담당자 정하기
5. 소통 채널과 방식 합의 ⇒ 어디서 이야기 하는지, 언제 이야기하는지, 기록은 어떻게 남길지

그래서 시작할때 정하면 좋은 질문들은

- 이 프로젝트는 무엇을 위한 프로젝트인지?
- 이슈가 생기면 누가 결정하는지?

기획이 정해진 후에는

- 꼭 있어야 할 기능(MVP)는 무엇인지?
- 없어도 되는 기능은 무엇인지?
- 언제까지 어느 수준으로 만들 것인지?



아이디어를 내는 것은 다같이 브레인스토밍을 하거나, 마케터 또는 PM이 시장조사를 하거나, 클라이언트가 수주하는 경우가 많아 따로 담지 않았습니다

서비스 구조를 그려보기

서비스 구조란 무엇일까요

단순하게 말하면 사용자가 버튼을 눌렀을 때, 무슨 일이 어떤 순서로 벌어지는지. 이 흐름을 눈으로 볼 수 있게 정리하는 것이 서비스 구조라 생각합니다.

로그인, 게시물 조회, 화상채팅 등 이런 기능들은 흐름의 결과기에 어떤 흐름이 존재하는지 먼저 파악 하는게 좋습니다

1. 사용자 화면 흐름부터 그리기. 단순히 종이에 그려도 좋습니다
(제일중요)
 - a. 사용자가 처음 보는 화면은?
 - b. 다음에 무엇을 누르는지?
 - c. 어디서 막히고 어떤 조건때문에 막히는건지?
2. 화면마다 필요한 데이터를 정리하기
 - a. 로그인 화면 → 아이디, 비밀번호
 - b. 목록 화면 → 해당 아이템에 대한 데이터
 - c. 상세 화면 → 단건 데이터
3. API 흐름을 연결하기
 - a. 이 화면은 어떤 API를 호출하는지
 - b. 언제 호출되는 건지
 - c. 성공/실패 시에는 어떻게 되는지
4. 데이터가 어디서 오는지 표시하기
 - a. 이 데이터는 DB에서 오는건지
 - b. 외부 API 호출로 발생되는건지
 - c. 계산의 결과일지

이렇게 하나하나 정해가며 불필요한 것은 줄이고, 해당 기능들을 왜 만들어야 하는지에 대해 명확하게 정해가기 시작합니다.

물론 완벽한 계획은 없습니다. 완벽할 필요는 없어요 중간에 달라지기도 하고, 잘못됐을 수도 있습니다. 그래도 확장성 있게 늘려가면 되니까 그려두는게 훨씬 도움되더라고요.

해당 과정을 했다면 어느정도 프로젝트의 윤곽이 나올겁니다

- IA(정보구조도)
- 와이어프레임

- 화면 ↔ API 간단 매핑