

Re:Camp Remember

Remember

최성민

BE



김영현

BE



안재진

BE

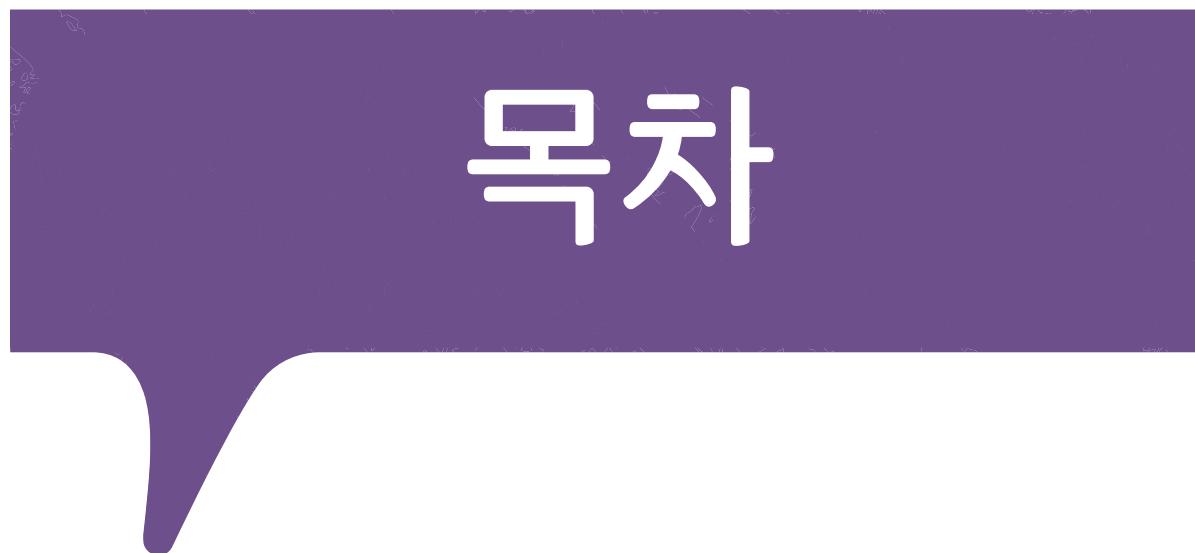


홍지현

BE



Remember



- 1 목표**
- 2 프로젝트**
- 3 관리**
- 4 계획**

Remember

개인 목표

Why Re:Camp?

What Is Goal?

How to work

팀 목표

윈터데브캠프 회고

What Is Goal?

How to work

Why Re:Camp?

Reason

윈터데브캠프 당시 미숙했던 모습이 아쉬움

기술에 몰두되어 기초에 관심이 없었음

머릿속의 내용을 문서화를 하지
않아 흐릿한 잔상만 남음

얻어가고 싶은 것

그 때보다 성장함을 스스로에게
증명 -> 자신감 회복

이번에는 기술보다 기초 중심으로
프로젝트를 진행해보고 싶다

프로젝트 종료 후에도 사고과정을
알 수 있을정도의 문서화

What is Goal?

Current

Code And Fix 방식으로 개발하여 수정사항 발생 시 어느 부분이 잘못되었는지 처음부터 찾음

기초는 홀대하고 MSA, Jenkins, Kafka, Docker 같은 특정 기술 스택에 관심을 가짐

계획이 없고 생각날 때마다 프로젝트를 진행하는 관리가 없는 방식

After

모든 개발 과정을 구조화 한 후 기록하기

단순 기술 구현 중심에서 기초 중심으로의 변화

철저한 계획과 관리를 통한 프로젝트 진행 경험하기

모든 개발 과정을 구조화 한 후 기록하기

1. Service, Repository, Package 구조의 변화, method 등을 모두 기록
2. 메소드, 변수명, Package 구조, Repository, Service의 사용 이유도 모두 기록
3. 퇴근 하기 전 개발일지 작성
4. 프로젝트 시작 전 DB Table, API 명세서 기록하기
 - > 무조건 캠프장님께 멘토링 받기, 시니어의 관점은 다를 수 있기 때문에
 - > DB Table, API 명세서 변경 사항 기록, 변경 이유도 기록하기

단순 기술 구현 중심에서 기초 중심으로의 변화

- I. 새로운 기술스택 사용 시 해당 기술과 다른 대체 기술,
스프링 자체 지원 라이브러리 등을 비교하여 문서 작성

- > 사용하는 이유를 잘 모르는데 제대로 사용할 수 없다고 느끼기 때문에
- > 단순 기술구현이 목표가 되어서는 안되기 때문에

2. 스프링 자체 지원 라이브러리 사용시 사용하는 이유 문서 작성

- > 예시) 스프링 시큐리티 대신 필터, 인터셉터와 AOP를 활용하여 인증 처리 구현 할 수 있는데
스프링 시큐리티를 사용하는 이유는?

3. Package 구조부터 비즈니스 로직 구조까지 설계한 이유 작성

- > 무조건 캠프장님께 멘토링 받기, 스스로 찾지 못하는 부족한 부분이 무조건 있기 때문에

4. Re:Camp 기간에 어떻게 약점을 보완할 수 있을까?

현재 스스로 가장 미흡하다고 생각하는 기초부분 -> DataBase

- > DataBase를 공부해본 적이 없으며 기초지식이 부족하다고 스스로 느끼기 때문
- > 단순 쿼리 작성도 힘든 상태, ORM인 Spring Data JPA로 해결해왔기 때문

- > 기초지식이 부족하기 때문에 매일 DB 학습 후 문서 정리, 코어타임 외적인 시간에 학습
- > DB뿐만 아니라 ORM 기술인 JPA도 학습 후 문서 정리
- > DB 성능 측정 후 Query 튜닝하면서 성능 향상 확인(ex. 정규화, 인덱싱)

철저한 계획과 관리를 통한 프로젝트 진행 경험하기

I. 팀원과 작성한 그라운드 룰, 프로젝트 관리보다 스스로 더 압박강도를 높여서 프로젝트 진행하기

- > 매일 코어타임을 초과하여 프로젝트 진행하기
- > 팀 프로젝트를 시작하기 전 맡은 파트 철저하게 분석하여 기능 구현 리스트 만들기
기능은 수정될 수는 있어도 추가되는 것은 최대한 지양해야 함

2. 계획을 수립할 때 계획을 세분화하고 수립한 이유 문서화하기

- > 예시) 단순히 언제까지 무엇을 하겠다가 아닌 이 사항은 이런 이유때문에 몇일이 소요될 것 같다
- > 예상한 소요시간과 차이가 발생할 시 차이가 발생한 이유도 문서화하기

What is Goal?

Reason

저번 캠프는 도전이 아닌, 할 줄
아는 것을 급하게 한 느낌

미흡한 설계와 개발

조건 내에서 충분히 어려우면서
새로운 도전 욕구

Effect

철저한 계획 하에 완료하는
프로젝트 경험, 성취감

조건

두 달의 시간 내에 해결 가능

경험해보지 않은 것

What is Goal?

Current

미흡한 설계에 기반한 개발

머리로만 생각하고 개발하여 추후 많은 부분이 변경되는 경우가 허다하다.

일정 관리의 미흡함

코드 변경이 잦으니 개발 기간도 늘어나는 경우가 많았다.

코드 작성 근거의 부족

기능에 대해 이유를 모르고 다들 하니까 똑같이 작성하는 습관을 가졌다.

After

실행 흐름에 기반한 코드 설계, 이를 바탕으로 한 개발하기

구현하려는 기능에 대해서 전체 흐름을 완전히 이해하고
추후 설계가 바뀌지 않는 개발을 할 수 있는 상태

사용하는 기술에 대해서 “왜?”라는 질문에 설명 가능 할 정도의 상태

실행 흐름에 기반한 코드 설계, 이를 바탕으로 한 개발하기

1. 기능의 전체 실행 흐름 작성 (예시 : 채팅)
2. 함수의 실행 흐름 작성 (예시 : 채팅 전송)

주기적인 기록과 관리로 개발 일정 관리하기

그라운드 룰에 의거한 백 로그, 회고 작성. 퇴근 전 개발 일정을 체크

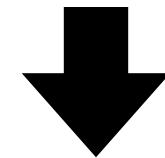
사용하는 기술에 대해서 “왜?”라는 질문에 설명 가능 할 정도의 능력 함양

설계 시 설명이 담긴 문서화, 코드 작성 시 주석 작성 등

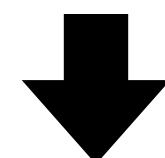
안재진

Why Re:Camp?

내가 캠프에 와서 프로젝트를 하는 이유가 뭘까?
성장하기 위해서



나의 약점을 채우면 성장하지 않을까?
그렇다면 나의 약점을 보완하자



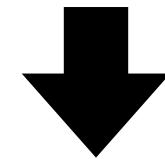
보완하고 싶은 약점은?
문제해결 능력

안재진

Why Re:Camp?

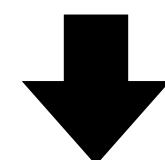
왜 문제해결 능력?

개발을 하면 여러 다양한 문제에 부딪치게 되고 그러한 문제를 효율적으로 해결하기 위해서는 본인만의 문제해결 능력이 필요하다고 생각



미흡한 이유는?

명확한 설계 없이 구현하기에만 급급했고 문제 해결 과정을 기록하지 않아 회고를 하기 위한 데이터를 쌓지 못함



그렇다면 문제해결 과정을 만들어 프로젝트에 적용하고 기록하여 회고 데이터를 쌓으면 나의 약점을 보완할 수 있지 않을까?

What is Goal?

Current

명확한 설계 없이 구현하기에만 급급한 상태

부실한 문제해결 과정

문제 해결 과정을 기록하지 않아 회고를
하기 위한 데이터를 쌓지 못함

After

기능 요구 사항 정리, 계층간 흐름을 파악한
기능 설계로 변화

문제해결 과정을 정리하고 프로젝트에 적용해
과정을 수정해 나가면서 나만의 문제해결 과정 정리하기

개발 과정 및 문제해결 내용을 문서화하여
회고 데이터 쌓기

I. 기반 기술에 대해 조사하기

기반 기술들에 대해 비교 조사하여 프로젝트와 적합한 기술 선택하기

2. 선택한 기술에 대해 기본지식 학습

기술 스택의 작동 원리를 파악하며 정확한 이유를 가지고 사용하기

3. 위 내용을 바탕으로 기능 요구 사항을 정리하고 설계하기

각 서버에서 필요로 하는 기능 요구 사항 정리하기

시퀀스 다이어그램을 그려 계층간 흐름을 파악하며 기능을 설계하기

계층을 세분화하고 각 계층에서 어떤 역할을 하는지 구체화하기

4. 기능 구현하기

앞 과정을 통해 설계 후 설계한 내용을 바탕으로 기능을 구현하기

5. 위 과정을 개발 일지에 기록하고 점검하기

매일 진행정도와 남은 일정을 체크해서 설정한 스프린트 계획에 도달하고 있는지 점검하기

Why Re:Camp?

리캠프를 도전하는 이유

프로젝트 관리에서 시간 관리를
제대로 하지 못해 미완성으로 끝남.

이전 캠프가 마음에 와닿았음.

오프라인 환경에서 조금 더 프로젝트에
집중해보고 싶다는 생각이 존재하였음.

현재 상태

팀 프로젝트 관리에서 시간 관리가 미흡함.

웹 프로젝트에 대한 지식이 미비한 상태

게임 서버 개발자로써의 성장을 원함.

Why Re:Camp?

개인 목표

게임 서버에 적용해볼 만한 웹 기술 적용해보기

설계를 기반으로 한 나의 한계를 경험해보기

도달하고자 하는 상태

게임 서버에서 웹 서버를 접목 시킬 수 있는
게임 서버 개발자

팀 프로젝트에서 철저한 설계로
한계치를 발휘할 수 있는 개발자

게임 서버에 적용해볼 만한 웹 기술 적용해보기

1. 단순히 게임 서버에 적용만 하는 것이 아닌 기반을 이해하여 개발하기
 - 웹 서버의 지식이 부족하기 때문에 이를 채우기 위함
2. MSA, WebRTC 같은 큰 기술이 아닌 웹 로우 레벨을 다룰 수 있는 실력 기르기
 - Unreal, C++ 과 같은 언어는 로우 레벨을 설정해야 하는 경우가 발생하기 때문
3. 이러한 기능들을 구현하는 과정을 세세하게 기록하여 스스로 다시 확인

설계를 기반으로한 나의 한계 경험하기

1. 일정을 널널하게 잡지 않고, 타이트하게 잡아서 매일매일 일정을 완료할 수 있도록 하기
 - 일정을 타이트하게 함으로써 노력할 수 밖에 없는 상황을 만들기
2. 항상 하루가 마무리 되었을 때, 일정이 마무리 되지 못했다면 왜 못했는지 개인 회고
 - 왜 실패했는지를 기록함으로써 잘못된 부분을 고쳐나가기 위해

김영현 소통이 없고 일정관리를 하지 못한 상태로 프로젝트를 끝낸 것이 아쉬웠다
프로젝트를 마무리한 후 코드만 있고 왜 이 코드를 작성했는지에 대한 문서들이 없어서 아쉬웠다
본인이 무엇을 해야하는지 구체적으로 알지 못한 채 프로젝트를 진행해서 아쉬웠다

안재진 처음 사용해야 하는 라이브러리에 너무 겁을 먹어 목표를 세우는데 있어서 소극적
프로젝트 실패 요인중 하나는 명확한 관리와 계획이 없어서라고 생각

최성민 저번 캠프는 시간에 쫓기듯 캠프를 진행해서 인지 이유 없이 그냥 작성한 코드들이 많았다.
그래서 인지 그냥 “할 줄 아는 것”을 혼자서 급하게 해 본 느낌이었다.

홍지현 서로의 실력 파악이 제대로 되지 못하여 일정을 정하고 이를 행하는 것에 큰 차이가 발생하였고,
결과적으로 시간 관리가 되지 않아 프로젝트가 미완성으로 마무리되었다.

팀 목표

Team Goal

목표 1

각자의 약점을 보완해
프로젝트를 마무리하기

목표 2

설계에 기반하여
프로젝트 완수하기

- I. 아키텍처, 시스템 구조 설계 뿐만 아니라 프로젝트까지 A-Z를 전부 핸들링 하는게 설계
2. PMP(Project Management Plan) 자체가 프로젝트 설계라고 생각
3. 무엇을 해야하고 왜 그것을 해야하는지 근거를 확실히 아는 것이 설계
4. 적절한 시간 자원 관리로 진행되는 프로젝트
5. 기능간 데이터 흐름을 파악한 로직 설계

어떻게 설계에 기반하여 프로젝트를 완수할 것인가?

김영현

1. 맑은 파트에 대해 왜 이 파트를 선택했으며 무엇을 해야하는지 전부 정리하여 기록
2. PMP에 작성한 관리와 계획을 중심으로 프로젝트를 진행
3. 기반 기술에 대해 정리하고 다른 기술과 비교하여 어떤 이점이 있는지 정리하며 설계

안재진

1. 기반 기술에 대해 조사하기
2. 선택한 기술에 대해 기본지식 학습
3. 위 내용을 바탕으로 기능 요구 사항을 정리하고 설계하기
4. 설계한 내용을 바탕으로 기능 구현하기
5. 위 과정을 개발 일지에 기록 하고 점검하기

어떻게 설계에 기반하여 프로젝트를 완수할 것인가?

최성민

- 미흡한 설계에 대해서는 개인 목표의 달성 방법과 마찬가지로 진행한다.
 - 전체 기능의 실행 흐름 작성, 이후 세부적으로 들어가며 다시 한번 흐름을 작성.
- Node.js만 사용한다는 단점은 팀원들과의 문서 및 코드 공유로 타 언어에 대한 이해도를 높힌다.

홍지현

- 팀원들과의 연동하는 부분에 있어서 서로의 능력치를 확인하여 일정을 조율하기
 - 팀 프로젝트에서는 나의 일정만이 중요한 것이 아닌 팀원들의 일정 관리도 중요하기 때문
 - 철저한 자료조사 및 설계를 통해 프로젝트 일정 도출

프로젝트

프로젝트 목차

목차

선택 동기

소개

얻을 수 있는 것

김영현

- 실시간 상태관리와 커뮤니티 서비스 특성 상 SELECT하는 경우가 많은데 데이터베이스를 깊게 공부할 수 있을 것 같아 관심이 생겼음

최성민

- 영상 통화, 스트리밍과 관련한 기능을 다뤄보고 싶다. 또한 Slack, Discord와 같은 플랫폼에서 다시 보기, 스냅 샷 남기기 등 업그레이드를 진행 해보고 싶다.

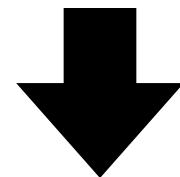
안재진

- 스트리밍, 채팅, 피드 등 여러 기능중에 평소 가장 많이 접하는 기능이 채팅이여서 기능들중 채팅에 가장 흥미가 생겼고 채팅 중심적인 플랫폼인 디스코드나 눈길이 감

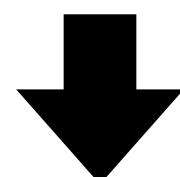
홍지현

- 웹을 개발해본 경험이 없어서 어떤 주제든 OK, 또한, 소켓을 주로 다루다 보니 도움이 될 부분이 많다고 생각하였음.
- 자주 사용하는 앱이여서 생각해볼 거리가 많다고 생각하였음.

팀 목표 적용에 서비스의 형태가 중요하진 않다



팀원 개인의 니즈를 만족시키는 서비스를 탐색



- 상태 관리, 채팅, 영상 등의 기능
- 프론트 개발자 부재로, 클론 가능한 디자인

하모니는 플랫폼 디스코드를 클론 한 프로젝트

하모니의 의미

팀원 한명이 이탈했기 때문에 디스코드의 원래 뜻인
불협화음에서 반대 뜻인 조화를 이뤄나가자는
뜻으로 팀 프로젝트 이름을 하모니로 결정

- 실시간 상태 관리 경험
- 데이터베이스 모델링 이해도 상승
- 미디어, 시그널링, 채팅 서버 등 새로운 경험
- 실시간 상호작용 기능 구현 경험
- API 게이트웨이 구현으로 로우 레벨 다뤄보기

목차

그라운드 룰

프로젝트 위험 관리

기타 관리

회의 종류

스프린트 작성 후, 완료 시점 정기 회의 개최(2번)

스프린트 회고 (매주 1회)

월요일 오후 4시

- 개발하면서 생겼던 이슈
- 개발하면서 부족하다고 느낀 부분
- 개발 외적으로 생겼던 이슈
- 실패한 부분과 그에 대한 방안 대책
(위험 관리에서 정한대로 진행)

데일리 스크럼(매일 10:10)

캠퍼스 1층

주요 회의 내용

정기 회의

매주 화요일 오전 10시 30분

본인의 스프린트 업무 계획을 공유.

데일리 스크럼 회의

- 전날 업무에 대한 간단한 회고
- 금일 업무 계획을 공유
- 팀원 컨디션 확인

규칙

- 회의를 할 때마다 회의록을 작성하여 기록
- 정기 회의는 1시간을 넘기지 않는다
- 스크럼 회의는 10~20분을 초과하지 않는다
- 점심 식사 다같이 먹기
- 메신저 읽었으면 반응하기
- 회의는 원할때마다 할 수 있다
- 코어타임 오전 10시 ~ 오후 5시
- 기본적인 예의 지키기

문서화

- 팀 노션 페이지에 본인이 했던 내용들 문서화하기
(다른 팀원이 보고 기능을 이해할 수 있을 정도로)
 - Daily 개발일지 작성
 - 개인 목표에 따른 개별 회고

태도와 책임감

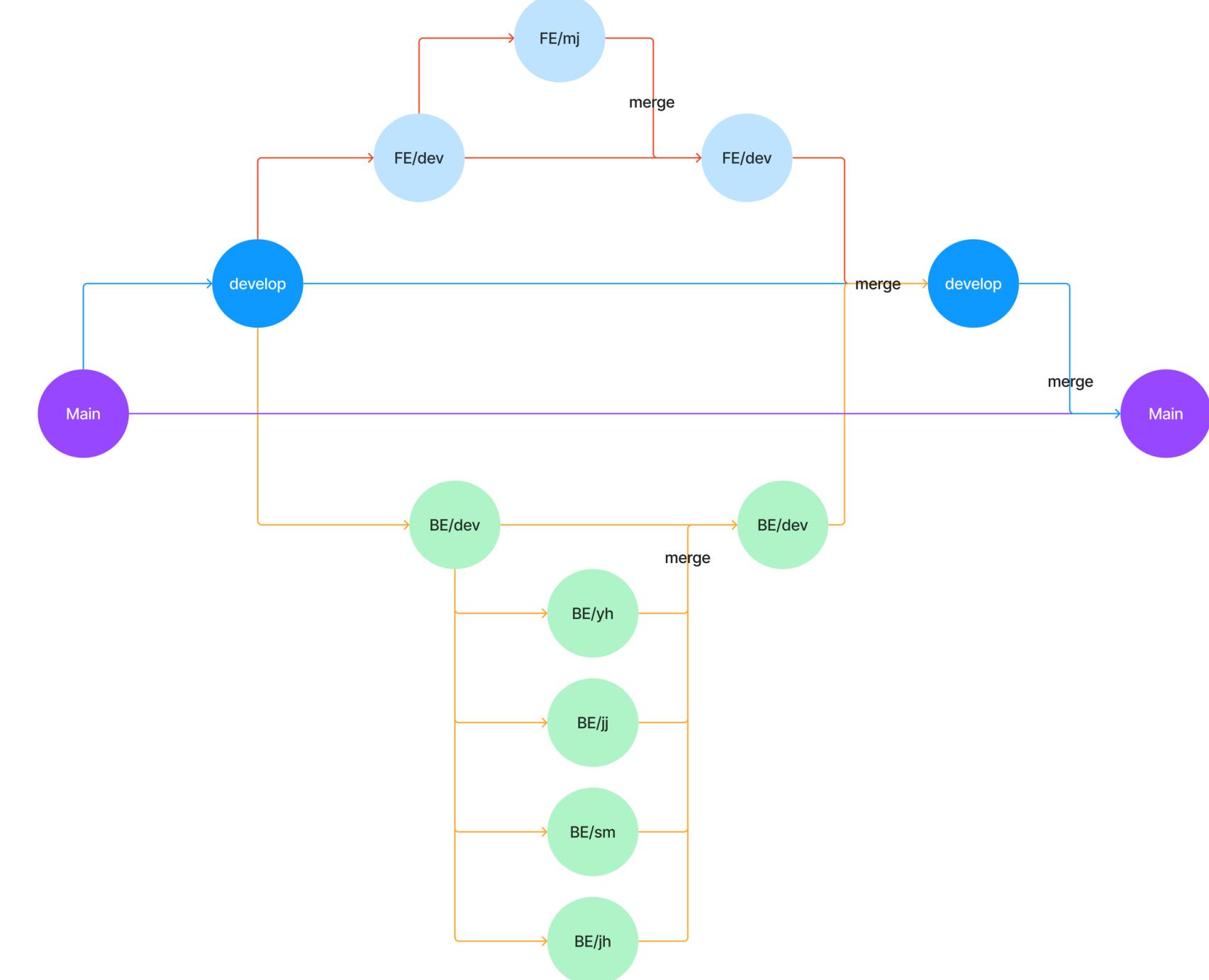
- 팀원이 이탈할 경우 팀에는 커다란 타격이 있다.
- 이탈하지 않더라도 불성실하게 참가할 경우 팀원들에게 피해를 주게 된다.
- 너무 의욕이 앞서서 강압적인 태도로 참여하는 것도 주변을 불편하게 만들 수 있다.
- 캠프에 참가하기로 결정했고, 팀 구성까지 완료된 후에는 팀원으로의 의무가 있다.
- 책임감을 가지고 활동하지 않으면 팀에 큰 피해가 있으니 이 부분을 꼭 염두에 두어야 한다.

코드 컨벤션 관리

- 어노테이션 짧은 → 긴 것
- 탭, 스페이스바

소스 코드 관리

- Github를 통해 프로젝트 소스 코드를 관리한다.
- 각 인원 별로 branch를 다르게 구성하여 develop branch로 merge한다.
- develop 및 main branch로 병합하는 경우 4명이 전부 동의할 시 merge한다.
- 깃 컨벤션 활용



본인의 스프린트 실패 했을 경우

계획한 스프린트 못 끝나는 경우

주차별 스프린트 못 끝내는 경우

스프린트 끝나기 3일전에 양심 고백하기(나 못 끝낼꺼 같음!)

어떤 부분이 문제인지 원인 파악하고 주변 사람에게 도움을 요청하기(팀원, 캠프장님)

그 다음 스프린트에서 못 끝내는 경우

스프린트 계획 회의 2일전 일요일에 얘기해서 팀원들과 기능 축소 회의

빨리 끝낸 경우

한 사람이 계속 빨리 끝날 경우 그 사람은
다음주차에서 해야할 스프린트 가져와서 하기

설 연휴 진행

- 2월 12일은 정기 회의 날로 다른 장소에서 오프라인으로 보기.

2월 4일까지 못 끝내는 경우

- 못 끝낸거 그대로 성능 테스트를 진행하지 않고 성능 테스트 기간을 일주일 뒤로 미루고 성능 테스트 기간동안 기능 구현에 집중

이탈/휴식자

- 2~3일 정도의 개인 일정은 합의 하에 파트 축소를 진행하기.
- 이탈자의 경우 지금까지 파트에 대한 기능 문서화를 착실하게 진행하기,

최종 발표까지 못 끝내는 경우

- 스프린트 정기 회의 날짜인 월요일까지 개발 진행
- 그 후에 최종 발표 준비함.
 - 2월 19일까지만 개발 진행하고 그 다음에 못하면 원인분석 및 최종 발표 준비

충돌 및 갈등 관리

- 의견을 나누는 과정에서 충돌이나 갈등이 발생 할 수 있다.
- 이러한점은 어떻게 관리하면 좋을까

필수 사항

- 싸워도 다시 받아들일 수 있는 자세를 가지기
- 비꼬지 않기
- 상대말 끝까지 경청하기

해결 방법

- 그랬구나 게임
- 심판은 나머지 2명이 다수결로

계획

계획

목차

업무 분담

스프린트 흐름

스프린트 세부 진행

계획

업무 분담

최성민

김영현

안재진

홍지현

시그널링 서버

커뮤니티 서버

채팅 서버

인프라

미디어 서버

상태관리 서버

상태관리 서버

유저 서버

알림 서버

MUST
HAVE

반드시 해야
하는 기능

SHOULD
HAVE

부가 적으로
해야 할 기능

COULD
HAVE

가능하면
할 기능

스프린트 회고

- 개발하면서 생겼던 이슈
- 개발하면서 부족하다고 느낀 부분
- 개발 외적으로 생겼던 이슈
- 실패한 부분과 그에 대한 방안 대책
(위험 관리에서 정한대로 진행)
- 오후 4시

스프린트 계획 설정

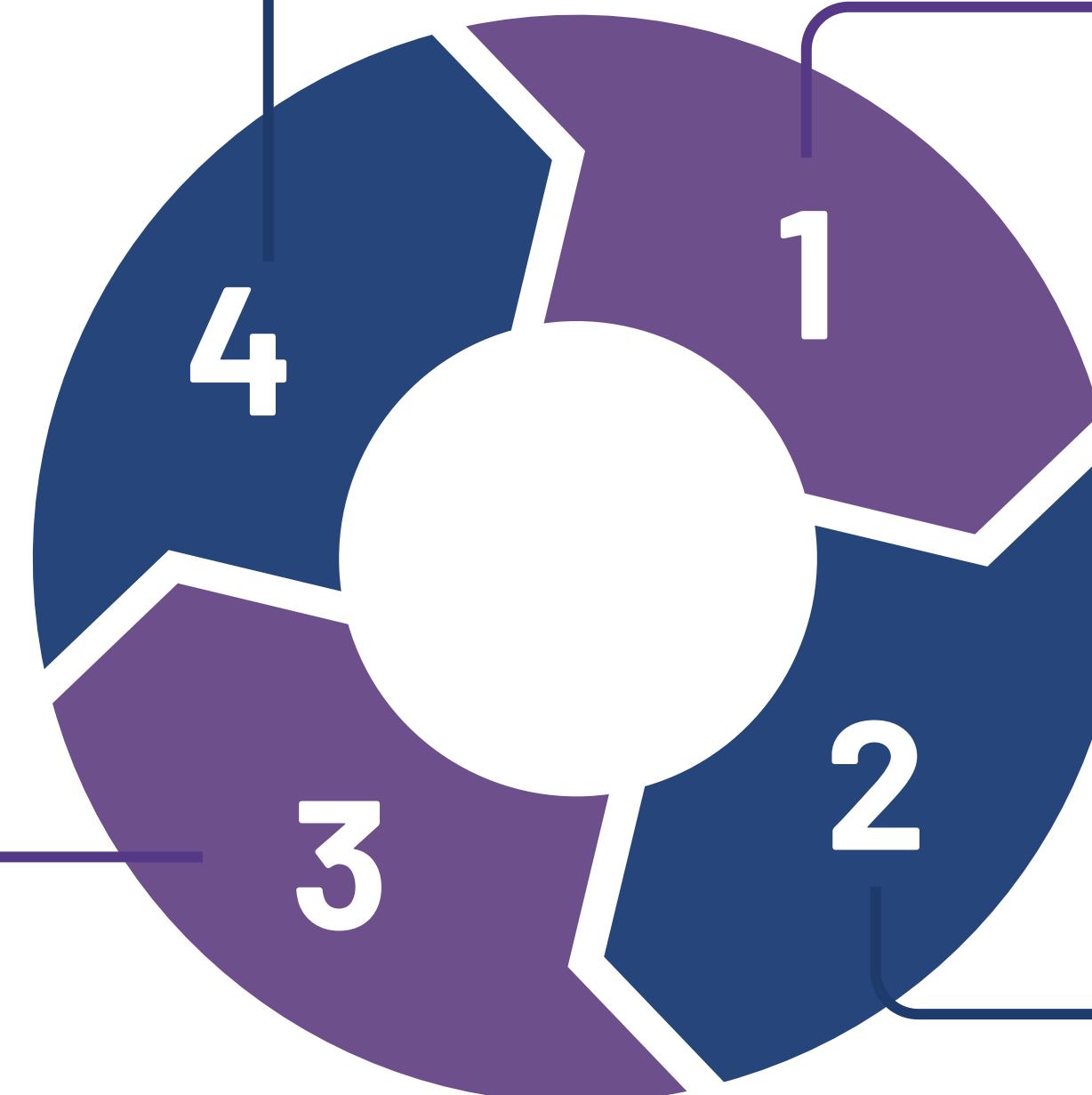
- 담당한 부분에 대해 계획을 작성하고 설명하기
- 전체 백로그에서 스프린트 1주 간격으로 나누기

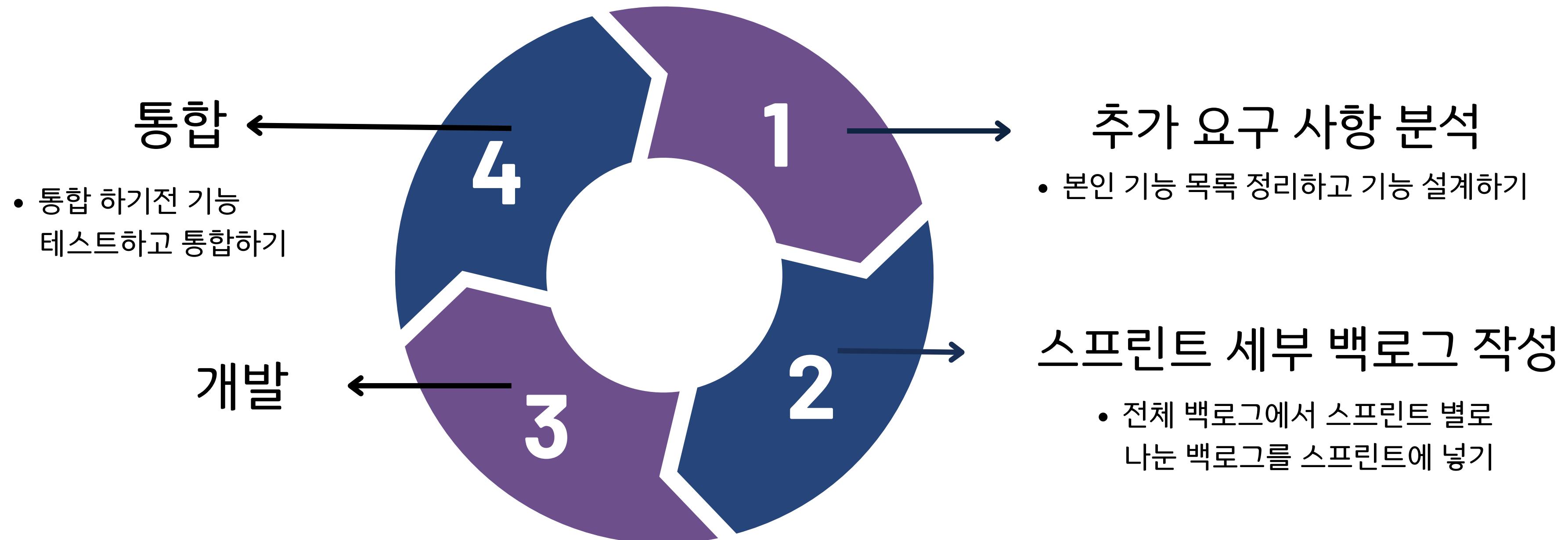
데일리 스크럼

- 전날 업무에 대한 간단한 회고
- 금일 업무 계획을 공유
- 본인 컨디션

스프린트 진행

뒷 페이지 작성





감사합니다!