



더 멋지게 사용하는 Git / GitHub

issue 템플릿 어떻게 만들지?



더 쉽게 사용할 수 있을까?

projects는 뭐지?

Git GUI Tool

git을 직관적으로 보면서 쉽게 관리하도록 도와줌

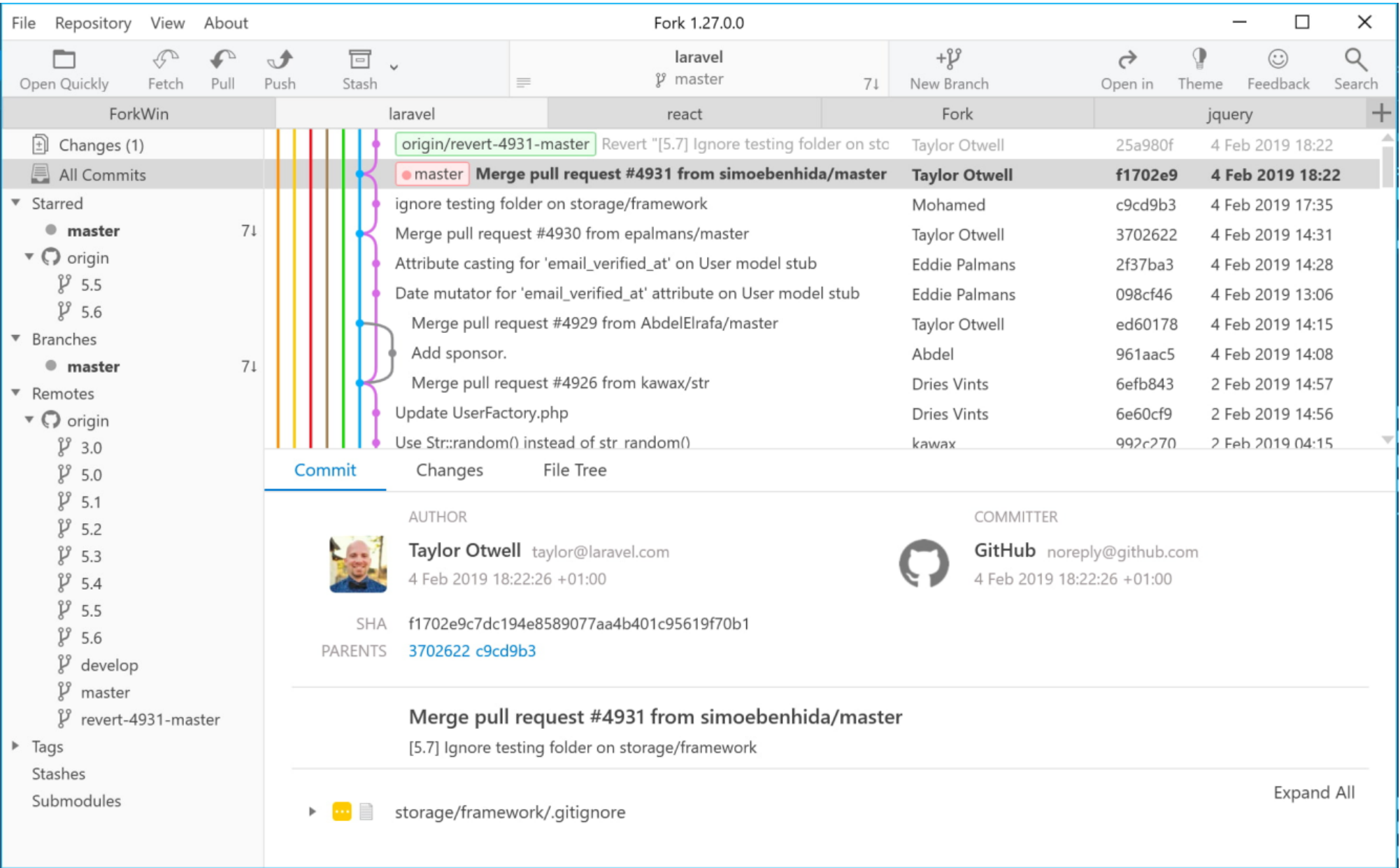


GitHub Desktop



Fork

Git GUI Tool



Fork

Github Copilot



더 빠르고 적은 노력으로 코드를 작성하는 데 도움이 되는 **AI 코딩 도우미**

언어만 지원하는 게 아니라 프레임워크도 지원

Github Copilot

```
function formatDate(date: Date): string {  
    return date.toISOString()  
}
```

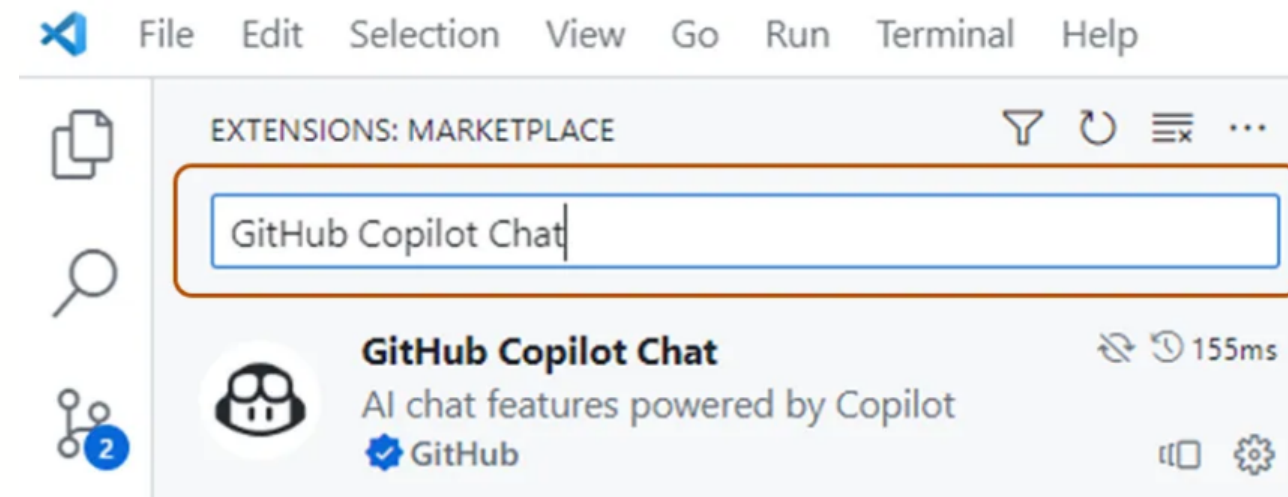
코드를 제안해주면 **Tab**을 눌러서 수락 가능

다른 제안을 보고 싶으면 **Alt +] (option +])**를 눌러서 다른 코드들을 볼 수 있음

여러 제안을 한 번에 보고 싶으면 **Control + Enter**

Github Copilot

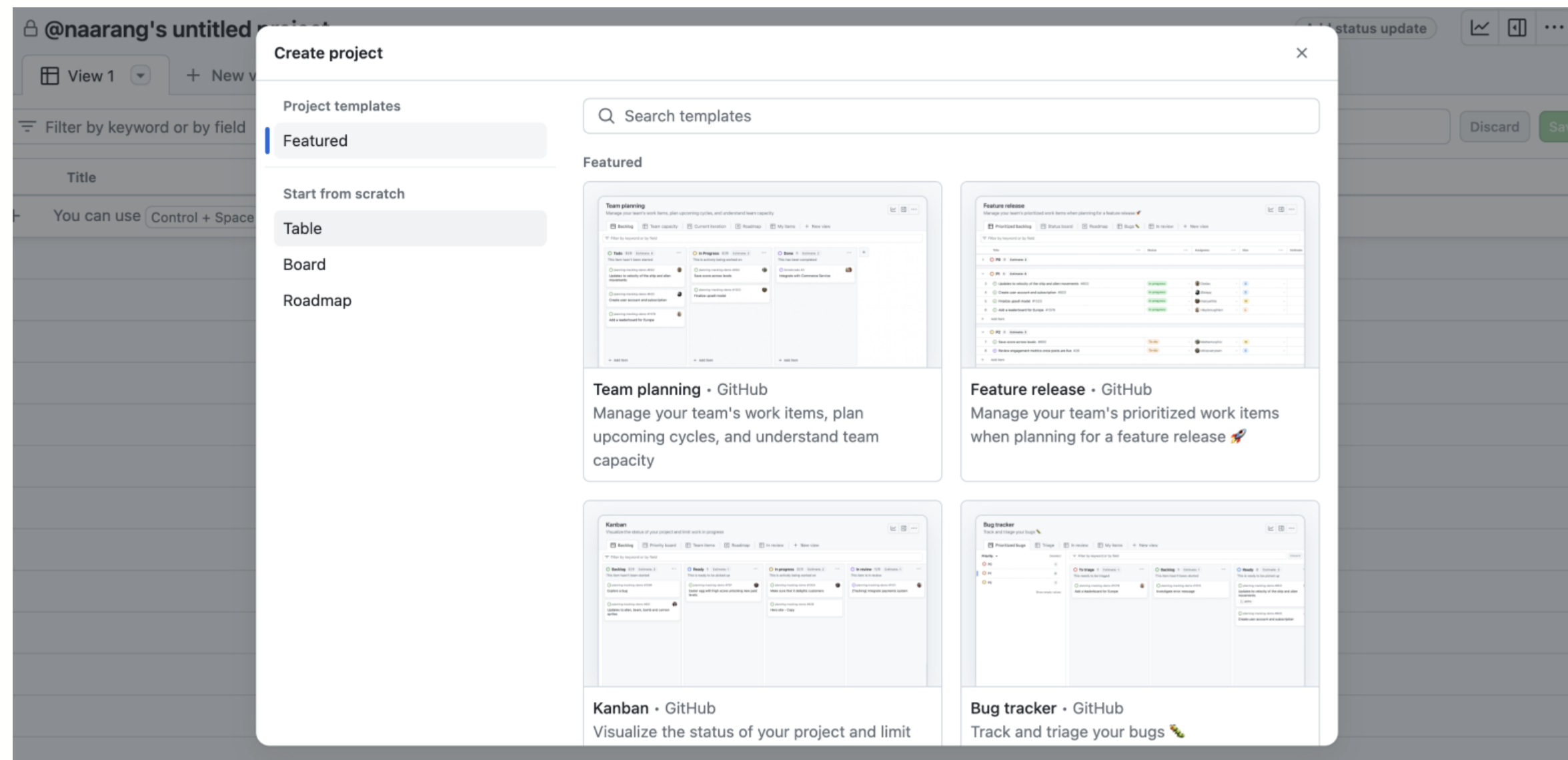
VS Code를 포함해 5개의 IDE를 지원 → 확장을 설치



GitHub Copilot Chat → 코드 작성 요청, 코딩 관련 질문 등이 가능

GitHub의 Projects

다양한 템플릿을 제공해서 프로젝트 관리를 도와줌

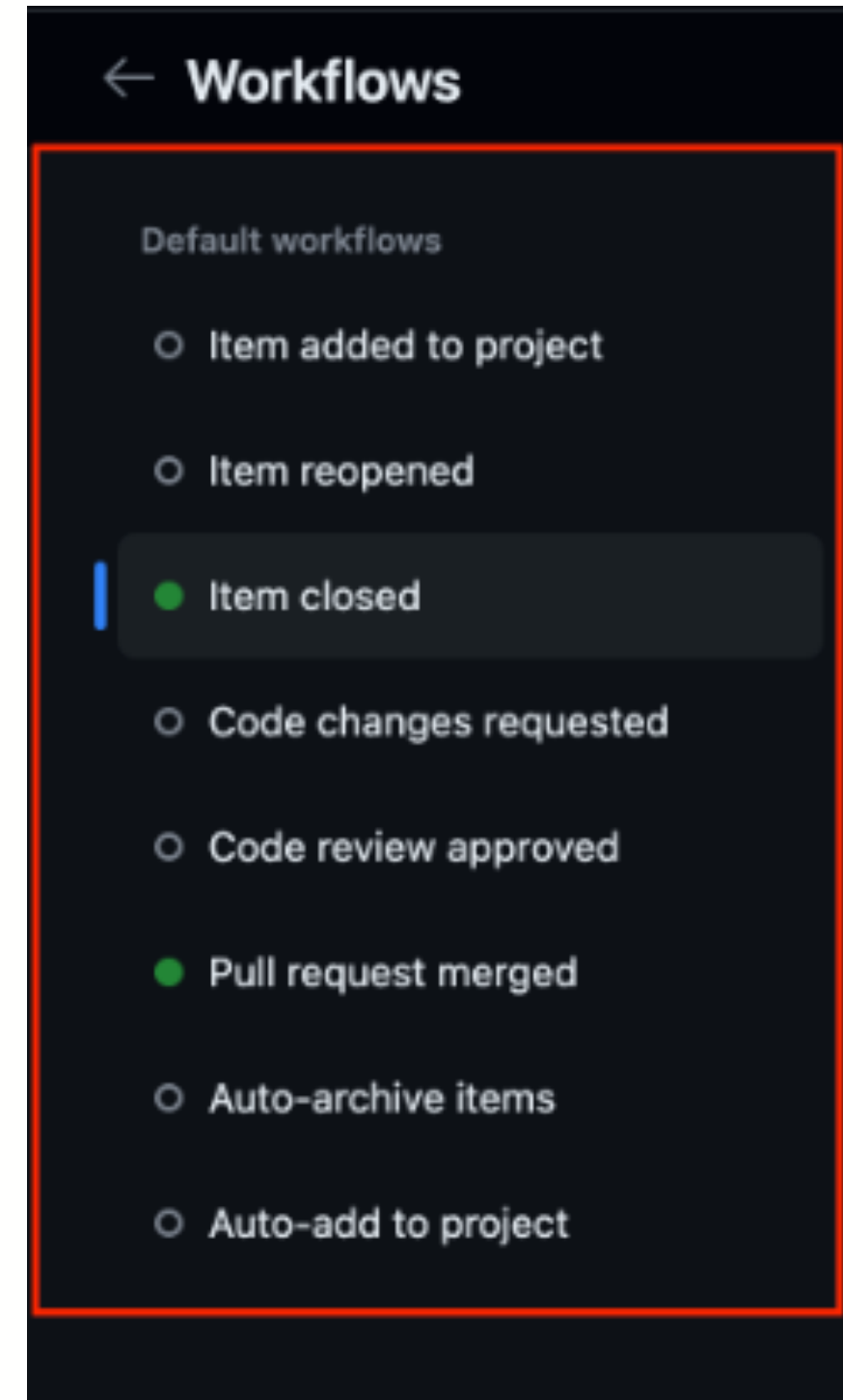


GitHub의 Projects

Workflow

필요한 작업을 설정하고 자동으로 작업 처리

Issue, pull request 관리 가능

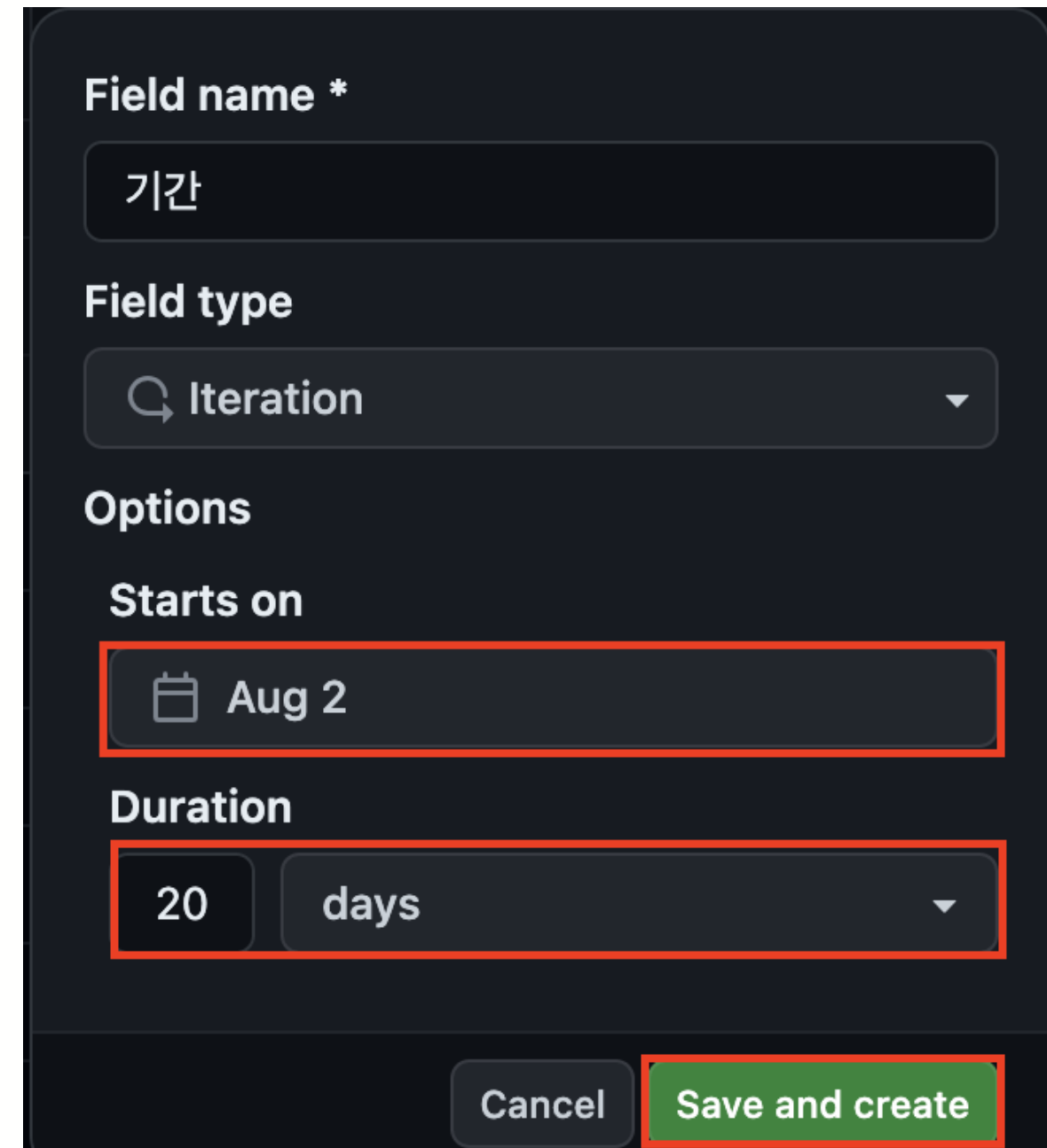


GitHub의 Projects

Iteration field

시작 시간과 지속 기간을 지정

→ 프로젝트의 작업별 세부 계획 관리 가능

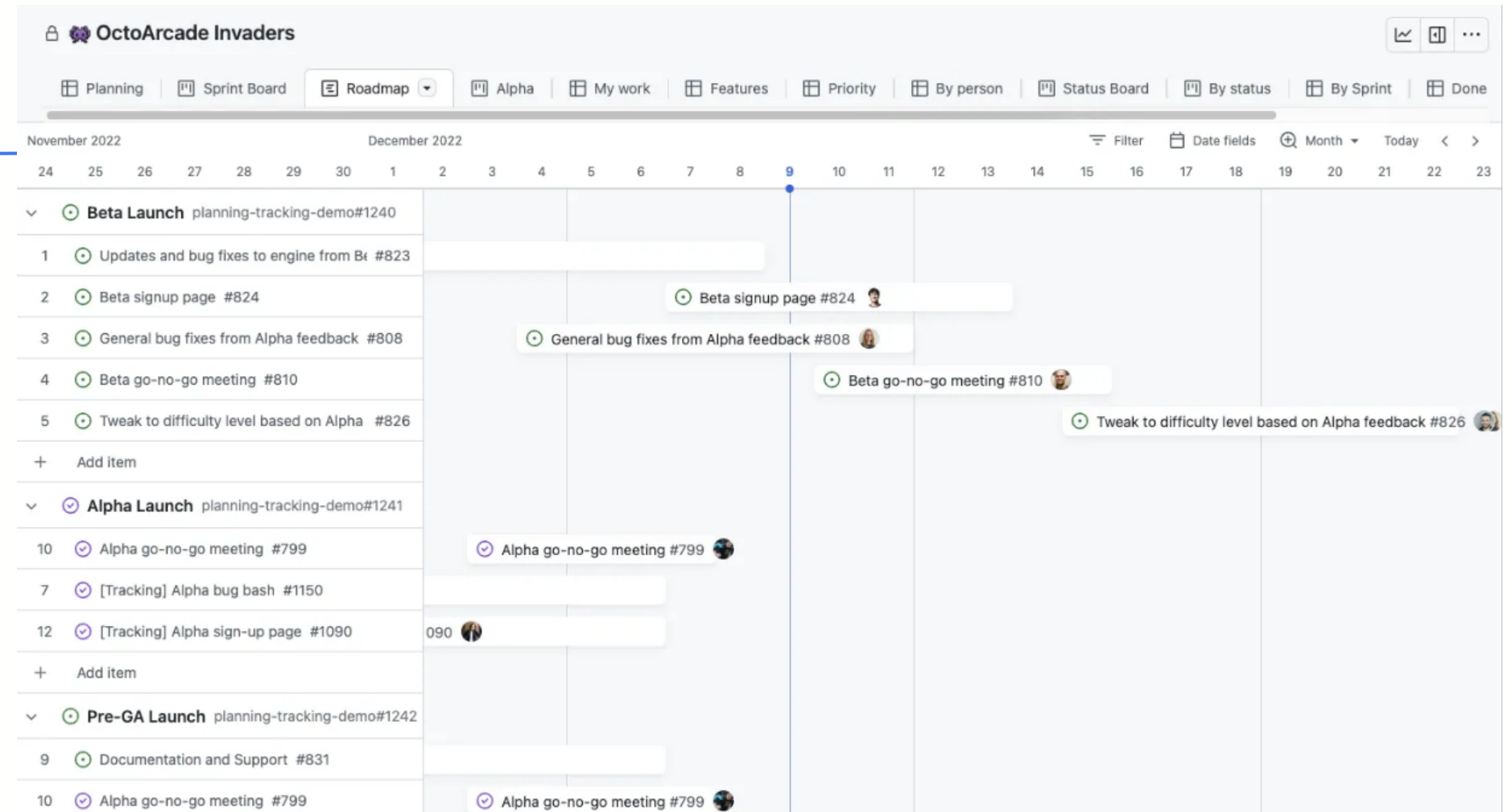


The screenshot shows the configuration interface for a new 'Iteration field' in a GitHub Project. The modal is dark-themed and contains the following elements:

- Field name ***: A text input field containing the Korean word '기간' (Period).
- Field type**: A dropdown menu with a circular arrow icon, currently set to 'Iteration'.
- Options**: A section containing two fields:
 - Starts on**: A date picker field showing 'Aug 2', highlighted with a red border.
 - Duration**: A field with a numeric input '20' and a unit dropdown set to 'days', also highlighted with a red border.
- Buttons**: At the bottom right, there are two buttons: 'Cancel' and 'Save and create'. The 'Save and create' button is green and highlighted with a red border.

GitHub의 Projects

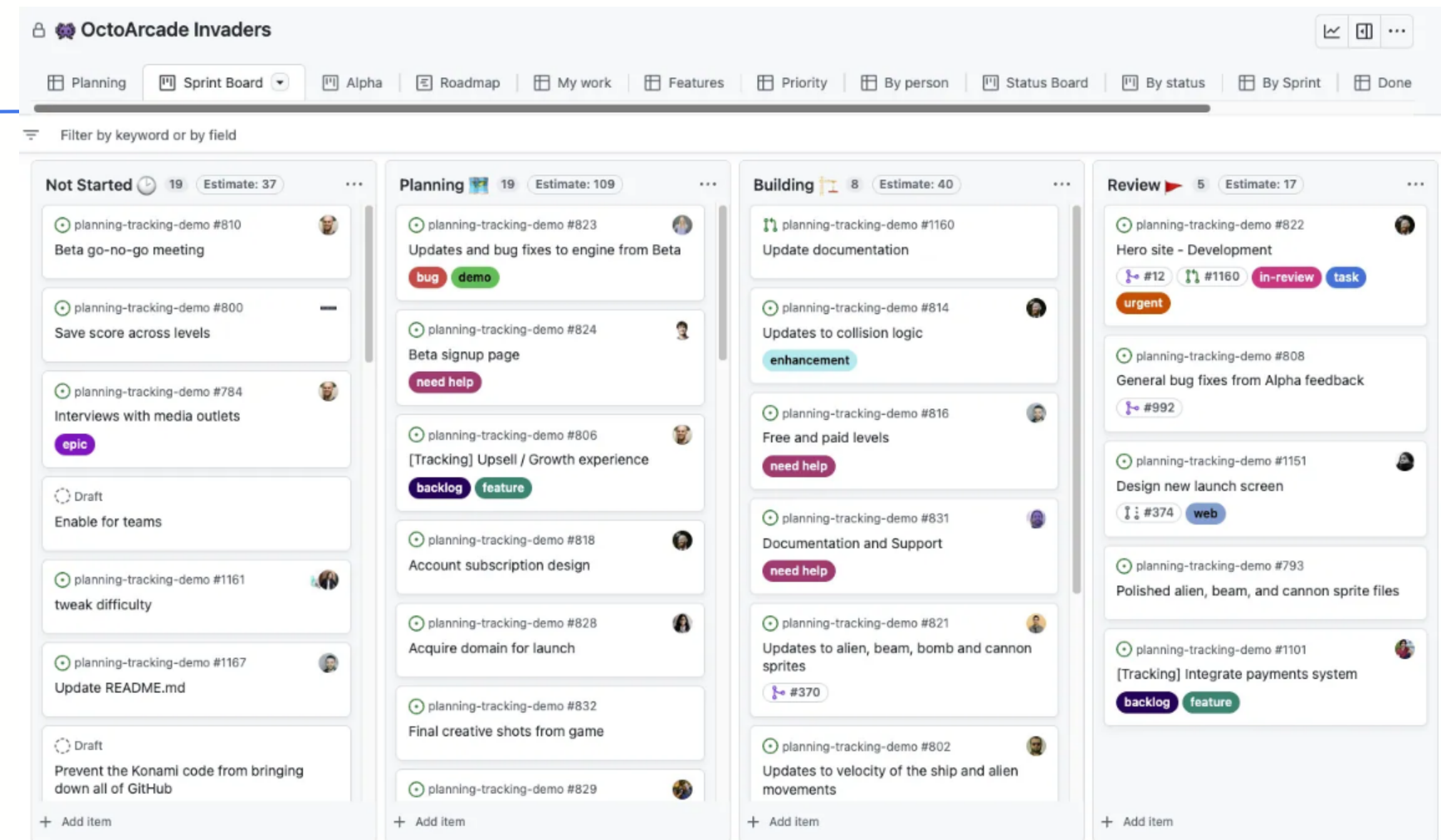
Iteration field



Roadmap 템플릿을 사용해서 기간별로 프로젝트 작업을 관리 가능

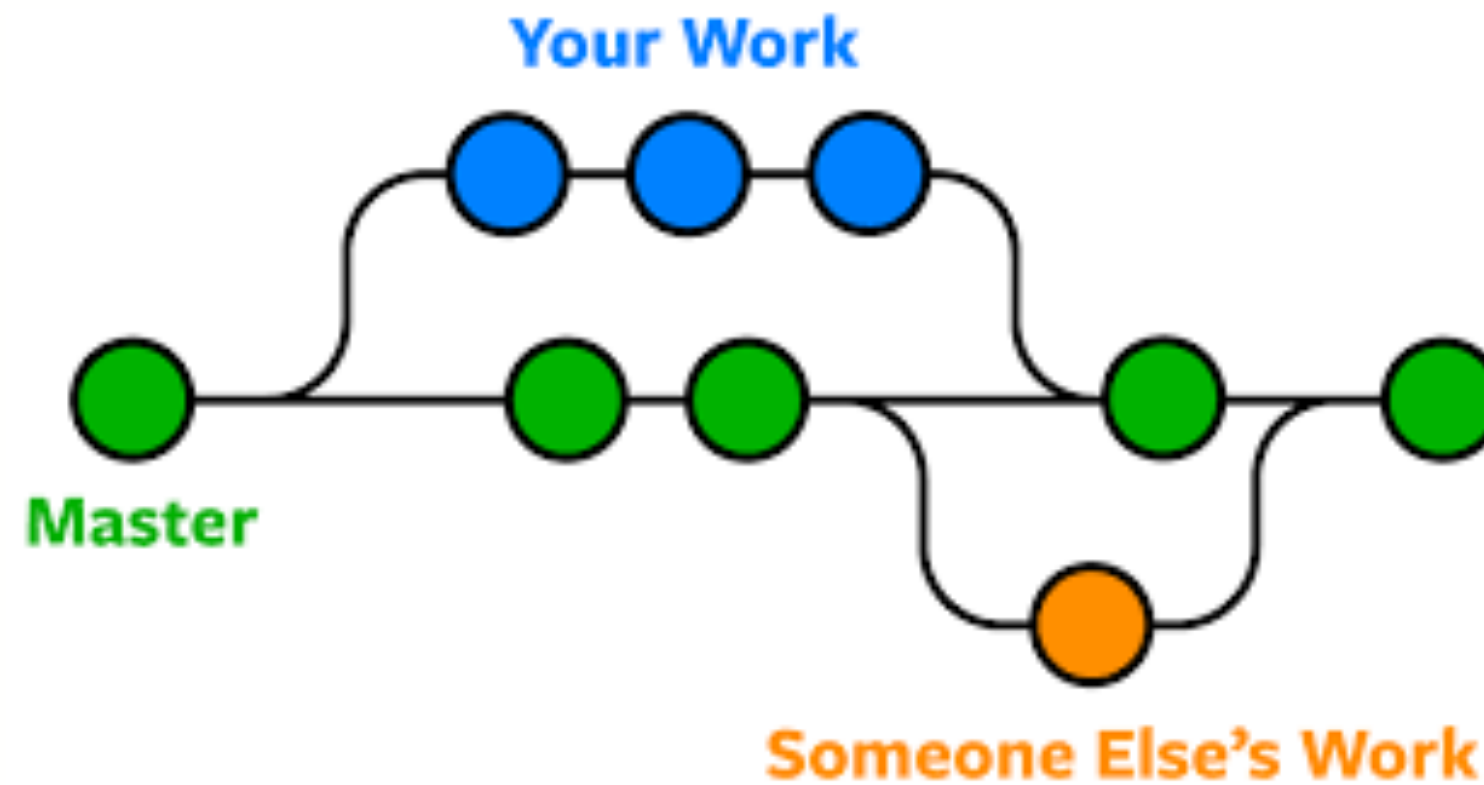
GitHub의 Projects

칸반 보드



칸반 보드 템플릿을 이용해 프로젝트의 업무량과 워크플로우를
더 효과적으로 시각화해 프로젝트를 관리 가능

Git 브랜치 전략



브랜치를 통해 여러 기능을 여러 사람이 병렬적으로 개발 가능

→ 브랜치를 효과적으로 관리

Git 브랜치 전략

Git Flow

Github Flow

Gitlab Flow

**Trunk-based
Development**

Git Flow



Main : 출시 가능한 코드

Develop : 다음 버전을 위해 개발 중인 코드

Git Flow



Release : 소프트웨어 배포를 준비

Hotfix : Main에서 발생한 문제를 바로 해결

Git Flow 단점

브랜치가 많이 존재하므로 여러 브랜치를 관리하기 복잡함
상대적으로 많은 변경사항을 한 번에 머지하게 됨

두 브랜치가 각각 독립적으로 오래 작업될수록
컨플릭트가 더 해결하기 어려워짐

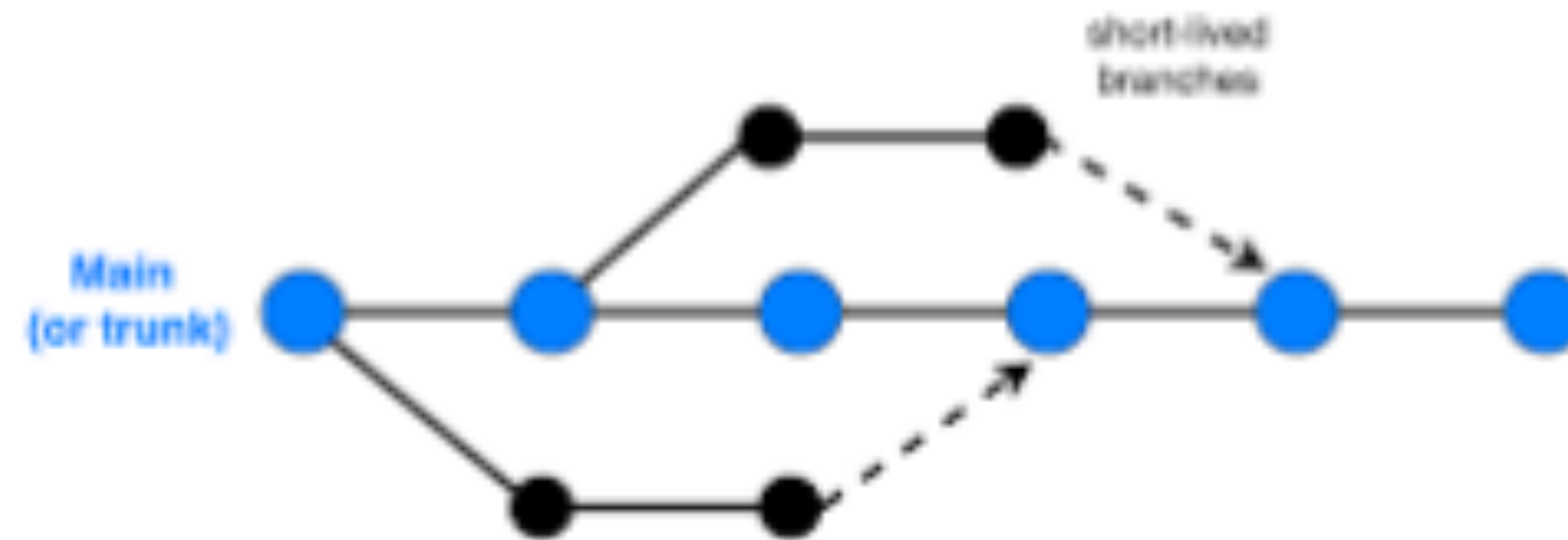
Git Flow 단점

Git flow는 웹 어플리케이션 개발에 적합하지 X

→ 스마트폰 어플리케이션, 오픈소스 라이브러리에 더 적합

웹 어플리케이션의 특성상 사용자는 항상
최신의 단일 버전을 사용하기 때문

Trunk-based Development

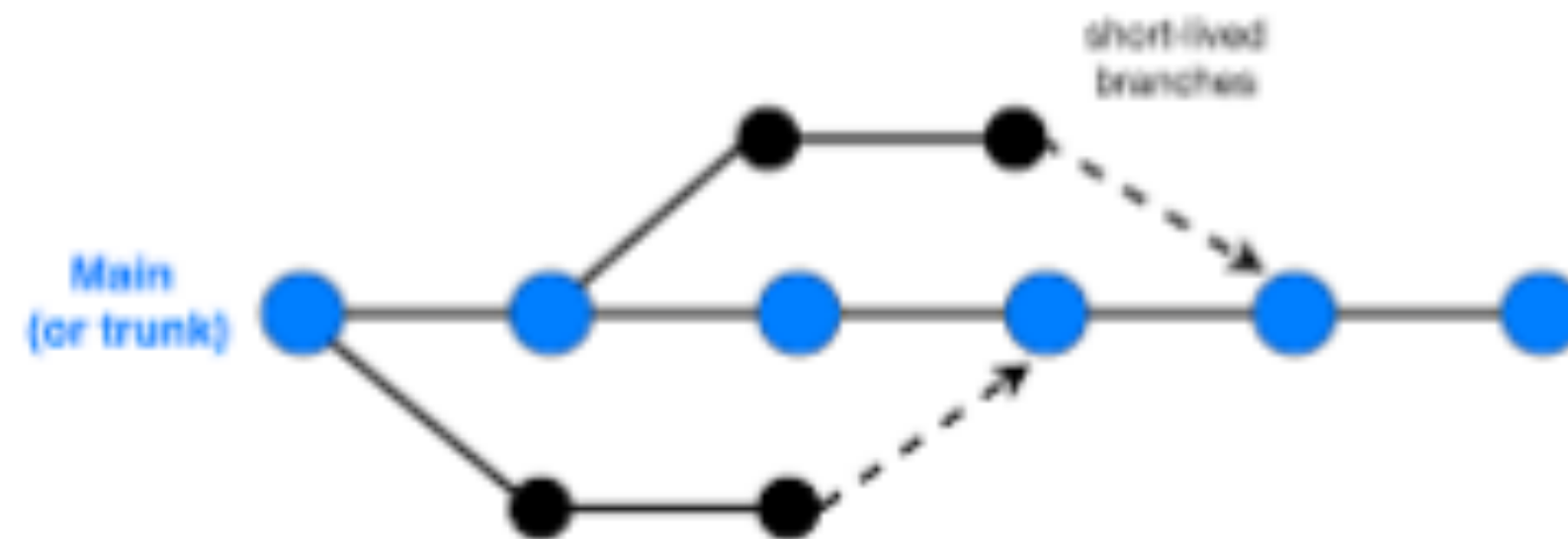


main(= trunk) 라는 주 브랜치 하나만을 운영

feature에서 기능을 개발 → main에 머지 → **자동화된 CI 시스템이**

main 브랜치에 대해 테스트 진행 → main의 코드 배포

Trunk-based Development



브랜치 관리 비용 감소

자주 머지하므로 컨플릭트 감소

배포 프로세스 간단화

THANK YOU FOR ATTENTION

이상으로 프레젠테이션을 마치겠습니다. 🔍