

개발 프로젝트 협업 툴

질문하기를 주저하지 말자! 내가 모르면 남도 모른다

DEV ENV 개발 환경

IDE, DOCS, ISSUES, VSC, CHAT

Developer Reference Check

평판 조회, 정말 한다!! ← 회사의 신중함 때문

Developer Onboarding

새로 입사한 사람들이 그 회사 문화, 시스템에 빨리 안착할 수 있도록 돕는 일련의 과정

부트캠프, 싸피의 계절학기

뉴비들이 빠르게 적응하기 위한 프로그램들!

특히나 코로나 시기인 지금 매우 중요하다

VPN

가상 사설 망, 회사 시스템에 접속할 때 사용

VDI

Virtual Desktop Infrastructure

가상으로 윈도우 하나를 지급받는 것과 같다.

보안성도 좋다.(DRM이 걸려있음 → 결제를 받은 다음 보안 해제 후 공유 가능)

IDE 통합 개발 환경

- VSC, Eclipse,
- 자동 완성, git graph, 버전 관리 등이 가능하다
- 테스트 케이스 호출이 가능해서 웹과 번갈아가며 볼 필요가 없다.

- 생산성이 올라간다!
- 디버깅이 편하다
- extension을 통한 확장성이 좋다.
- 다른 사람들이 쓰는 것도 보면서 배우자~~

VCS (Version Control System)

- 예전엔 SVM(중앙에서 관리)
- GitLab, GitHub, Bitbucket
하나가 익숙해지면 다른 툴도 잘 사용할 수 있다!

Issues 이슈/프로젝트 관리

- 예전엔 Excel이용 : 버전이 달라지고 수동으로 해야하므로 불편
- JIRA, Trello, REDMINE
 - JIRA의 경우 자유도가 매우 높다! (customize 가능! -ex. code review 단계 추가)
현업에서도 쓰임
개발자 이외로 다른 사람들도 함께 쓰기에 좋음

DOCS 공유문서 관리

- wikipedia : 집단 지성 good, 권한 설정 불가
- Confluence : 설치형, 회사에서 쓰임
- Google Docs
- Notion : 문서를 함께 편집할 때 좋음

CHAT 커뮤니케이션

- 카카오톡
- JANDI : 회사에 특화됨 - 조직도 확인 가능, 전자 결제 가능
- slack : 개발자들이 많이 씬 - 다른 툴과 확장성이 좋음

Integration

- Mattermost : Git lab, Webex, Google Docs, JIRA 등과 연동 가능

정리!

- 하나를 깊게 알면 비슷한 도구도 쉽게 적응 가능
- 개발자 코딩 실력과 도구를 다루는 실력 사이 밸런스 잡기
- 좋은 도구를 “효과적으로” 통합하면 프로젝트 가시성&활용성이 좋아진다
- 원격으로 협업할 때는 도구를 활용한 활발하고 적극적인 커뮤니케이션 노력이 필요하다.