

SW마에스트로 제13기 프로젝트기획서

1. 프로젝트개요서

프로젝트명	DORA Metrics를 이용한 개발 생산성 향상		
프로젝트 요약	팀 단위에 개발에서 이루어지는 협업 과정에서의 생산성과 안정성을 재고하기 위해 Google의 보고서에서 제시된 DORA Metrics를 기반으로 팀의 현상태를 분석하고 시각화해서 제시하고, 이를 토대로 팀의 발전 방향을 제시하는 서비스이다.		
기술 키워드	WEB/실시간 통신/Statistical Analysis		
ICT연구개발 기술 분류	①(대분류) 웹 애플리케이션 - ②(중분류) 생산성 툴, 애자일		
팀 명	알락꼬리여우원숭이	팀 원	류동인, 조인혁, 백종현
목적 및 필요성	1. 팀의 현상향을 객관적으로 진단하고 수치화 시키기 위해서 너무 많은 시간과 비용이 발생한다. 2. 내가 하고있는 노력이 팀의 성장에 얼마나 기여하고 있는지 시각적으로(또는 수치적으로) 알 수 없다. 따라서 효능감이 떨어진다. 3. 팀이 성장해온 과거의 기록을 볼 수 없다. 객관적 평가를 위해, 팀의 진행상황을 지표화, 수치화, 도식화를 통해 가시화 시키고, 생산성 관련 DORA metrics 기법을 이용하여 생산성을 향상시킨다.		
프로젝트 개요	팀 개발 진행에 있어 생산성 제고 및 지속 가능한 개발을 위한 방법론인 Dora Metrics를 이용한 프로젝트를 진행할 예정이다. 프로젝트 예상 기능 - Team Dora Metrics 지표 추출 및 제공(GitHub, BitBucket, GitLab) - 개발자 개인 역량개발 습관 지표 제공 - 다양한 형태의 팀을 위해 Development Cycle 지표 수집 방식 Custom - 개발 중 Team에서 발생할 수 있는 개발 이슈들에 대한 위험 알림		
수행 방법 및 추진 일정	6 : 최종 목표 서비스 기획, scalable한 인프라 아키텍처 설계, scalable 한 ERD 및 객체 설계, UX/UI 디자인, 디자인 외주 및 확정 6~7(~15) : Github와 프로젝트의 연동을 통한 Dora metrics 생산성 지표 추출 서비스 7~8(~15) : 개발자 개인 역량/개발 습관 지표 제공 서비스 개발 8(15~) : 소프트웨어 마에스트로 내부 테스트 및 feedback 수집 9~10 : 팀 당 위험 알림 서비스, Development cycle custom 제작, feedback 을 통한 추가 기획 수정 관련 개발 11(~15) : 프로젝트 마무리 / 추후 성장을 위한 개발자 커뮤니티/소프트웨어 마에스트로 내부 테스트를 통한 feedback 수집		
기대 효과 및 활용 방안	리더 :개발 전반의 문제점을 감지하고 팀 분위기 반전 PM/스크럼 마스터 : 프로젝트 진행에 걸림돌이 되는 요소를 식별, 제거 상부 보고에 있어 객관적인 지표 획득 개발자 : 자신 및 팀원들의 상황을 파악, 개발 습관을 개선 팀 전체 : 개발 팀 전체의 효율성을 제고하여 높은 생산성 획득		
결과물 형태 및 서비스 방식	Web Application - 권한에 따라 팀 현황을 볼 수 있는 것이 달라지는 시스템 - 유저 레벨에 따라 볼 수 있는 회사/팀/구성원 레포트 계층적 레벨UI 노출 - 추가적으로 , Slack/Email을 통해 위험 요소 탐지 알림 - 개발자 개인의 개발 습관 파악 및 역량 인지를 위한UI		

2. 프로젝트 기획서

□ 목적 및 필요성

○ 문제인식

- 객관적인 팀의 문제를 진단할 수 있는 툴이 없다.
- 내가 하고 있는 노력이 팀의 성장에 얼마나 기여하고 있는지 시각적으로(또는 수치적으로)알 수 없다.
- 팀을 객관적으로 진단하고 수치화 시키기 위해서 너무 많은 시간과 비용이 발생한다.
- 팀이 성장해온 과거의 기록을 볼 수 없다.

○ 기획 의도(문제해결)

- 2021 Accelerate State of DevOps Report(Google Cloud DORA, 2021)는 생산성을 평가할 수 있는 객관적인 지표 DORA Metrics를 제안한다. 우리는 이를 이용해서 팀의 현 상황을 분석하고 제시하며, 생산성을 재고할 수 있도록 하는 툴을 제안한다.
- DORA Metrics 이외에도 Back Log 분석, 코드 분석과 같은 개인과 팀의 생산성을 평가할 수 있는 요소를 추가적으로 배치할 수 있게 한다.
- 버전 관리 툴(GitHub, GitLab, BitBucket), 프로젝트 관리 툴(jira, proofHub, teamwork, scorio)과 연결하여 개인과 팀의 생산성을 분석하고, 이를 시각화해서 보여준다.
- 커뮤니케이션 툴(slack, mail)과 연결하여 다양한 문제를 미리 예측, 식별하고 이를 알려준다.
- 선택한 기간에따라서 평가요소(DORA Metrics, branch 개수, 코드 품질)의 변화를 시각화 해주며, 서로 다른 기간의 분석 결과를 비교해 준다.

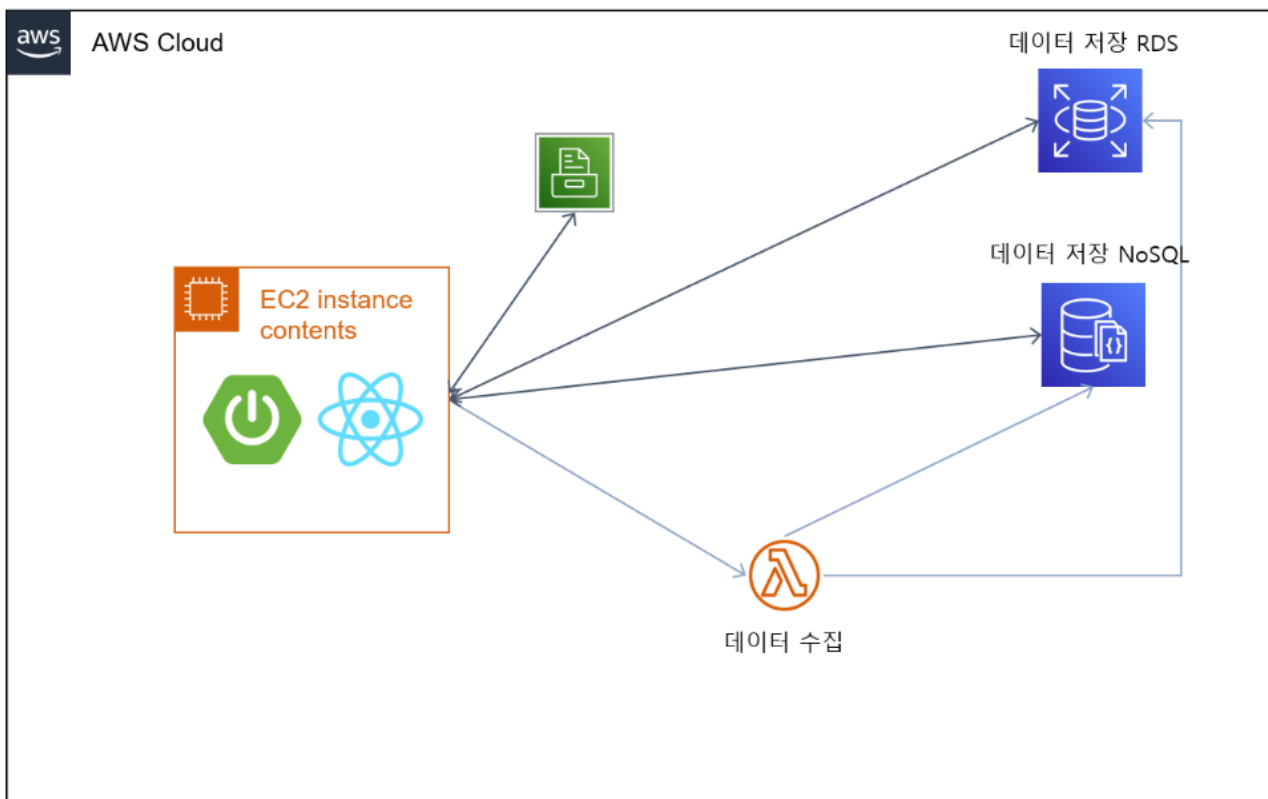
○ 시장/소비자 동향/분석, 경쟁제품 비교/차별화 요소

- 잡코리아X알바몬 통계센터의 조사에 따르면 성인 남녀 64%는 스스로를 '업글인간'이라고 여긴다. 특히 성인남녀 83.9%는 향후 업글인간 트렌드가 더 확산될 것으로 전망한다.
- 경쟁 제품들
 - Atlassian(Jira)
 - Jira의 보고서항목이 팀의 성과를 분석해서 제공한다. 하지만 백로그, 스프린트와 같은 것을 분석하는 방식이므로, Git을 이용한 DORAMetrics보다 세부적인 파악이 힘들다.
 - AzureDevOps Service
 - Jira와 같은프로젝트 관리 서비스를 제공하지만, 분석보다는 다양한 툴(Cloud, Git 등)의 통합성에 초점을 맞추고 있다.
 - GoogleCloud DevOps Solutions
 - Fourkeys : DORA Metrics를 측정해서 시각화해주는 DashBoard (결과를 분석해서 인사이트를 제공하지 않음)
 - 팀이 담당하는서비스의 자동화, CD/CI 등 기술적인 문제에 대한 솔루션을 제공
 - LinearB
 - 분석 없이 데이터 수집 후, 뿌려주는 것이 전부이다.
 - 직관적인 판단은 불가능한 상태로 뿌려 주기때문에 가독성이 떨어진다.
 - 번아웃이나, 한 번에 너무 많은 코드를 커밋하는 것 등 팀의 안정성에 대한분석은없다.
 - HayStack
 - 인사 관리를목적으로 하는 전체 솔루션의 일부로 Dora Metrics를 차용하는 형태이다. 따라서 단지 팀의 상태를 시각화 하는 것을 목적으로 한다.

- 차별점
 - 팀이 아닌 개발자 중심의 서비스
 - 기존 팀 기반 서비스에서 개발자 중심의 서비스.
 - 개발자 스스로 개발 성향 및 약점을 파악할 수 있도록 지금까지의 커리어에서 진행한 코드 습관의 통계를 보여준다.
 - 서비스 자체 랭크 서비스를 통해 자신의 현재 위치가 어디인지, 어떤 점을 개선할 수 있을지 보여줌으로써 더 나은 개발을 도모하게 한다.
 - 다양한 형태를 가진 팀들을 위한 솔루션
 - 개인과 팀의 성과를 다양한 형태로 분석할 수 있는 커스터마이징 기능을 제공한다.
 - GitHub뿐만이 아닌, GitLab/BitBucket 등의 Private Repository를 사용하는 회사에서도 사용이 용이하도록 한다.
 - GitHub 또는 Organization의 프로필의 ReadMe.md에 팀의 랭크를 알 수 있는 API 제공.

□ 프로젝트 개요

- 프로젝트 소개
 - 2021 Accelerate State of DevOps Report(Google Cloud DORA, 2021)에 등장하는 생산성이 높은 팀의 git 사용 방식 및 개발 단계를 분석한 DORA Metrics를 이용해서 팀과 개인의 높은 생산성과 안정성을 도모.
 - 다양한 형태와 상황의 팀이 사용할 수 있는 커스터마이징 가능한 팀 성과 분석 서비스.
 - 개발팀이 팀의 성장을 방해하는 병목을 객관적으로 파악하고, 이를 해결해 나갈 수 있도록 한다.
 - 개발팀이 마주할 수 있는 문제를 미리 예측하고 예방하며 관리한다.
- 시스템 구성도



○ 적용기술

구 분		항 목	세부 내용
S/W 개발환경	OS	Ubuntu 18.04(SVR)	Ubuntu 18.04 LTS(SVR) (~2023.04.27)
	개발도구	IntelliJ, VsCode, DataGrip	IntelliJ - JAVA Spring 개발용 VsCode - 기타 코드 개발용 DataGrip - DataBase 관리 및 ERD 설계
	개발언어	JAVA, JS, TS, SQL, Python, git shell script	JAVA - Spring 개발 JS / TS - FE (React) 개발 SQL - 프로젝트 쿼리 Python - 데이터 처리 Git shell script - GitHub 데이터 처리
기타(기자재 등)		AWS, MySQL, Redis, Docker	서버구성용, DataBase(MySQL, Redis), Virtual Server Hosting (Docker)

○ 주요 기능

- 버전관리 서비스(GitHub/GitLab/Bit Bucket)를 기반으로 Dora Metrics의 각 요소를 팀/개인에 대해 계산하고 시각화.
- 개발 중 발생할 수 있는 문제들(WIP, Too Short Review Time, Big Diff, Burnout possibility 등)을 자동으로 파악, 알람을 보여줌으로써, 놓칠 수 있는 개발 이슈들을 파악할 수 있도록 하는 기능.
- 서비스 사용 시점부터, 현재까지 개발팀에서 발생했던 문제와 팀의 생산성 평가 요소 변화를 보여주며 성장할 수 있는 방향성 제시.
- DORAMetrics 이외에도 backlog 분석, 코드 분석(클린 코드 기반 - Method 길이, 인자 수 등)와 같은 다양한 분석 도구를 지원하고, 이를 개인/팀 보드에 자유롭게 배치, 커스터마이징 할 수 있도록 함.

○ 개발 사양

- 개발 및 테스트 서버- AWS EC2, RDBS

□ 수행 방법 및 추진 일정

○ 주요 기능별 수행방법

Github 프로젝트 연동 데이터 추출	<ul style="list-style-type: none"> • Team Git accessKey 등록 후, git Organization 정보 검색 git script를 적용, 주기적인 정보 수집. • 더욱 상세한 정보를 파악하기 위해, selenium을 통한 크롤링
Team Dora Metrics 지표 추출 및 제공	<ul style="list-style-type: none"> • 추출 데이터 Database 저장 후, Restful API 구조로 서비스
개발자 개인 역량개발 습관 지표	<ul style="list-style-type: none"> • 추출 데이터 Database 저장 후, Restful API 구조로 서비스
Development Cycle custom	<ul style="list-style-type: none"> • Adaptable pattern 사용. 팀 형태에 따른 Cycle 판단 선택지 제공.
개발 위험 알람	<ul style="list-style-type: none"> • 주기적인 Team Git 수집 후, Slack Open API를 통해 알람

○ 수행방법 확보방안(아웃소싱 등)

- 디자인 아웃소싱을 통한 UI 디자인 개발.
- 주기적인 FE(JS/React) 개발스터디를 통해 FE 개발능력 향상
- 서비스 환경 구축이전, Docker Image를 통한 개발. 패키지 및 OS 버전 및 종류 차이로 인한 문제들을 최소화.

○ 추진 일정

구 분	추진 내용	추진 일정					
		6월	7월	8월	9월	10월	11월
계획	서비스 기획 및 개발 상세 로드맵 생성						
분석	서비스 필수 사항 확인 및 API 설계						
설계	최종 목표 서비스 기획						
	scaleable한 인프라 아키텍처 설계						
	scaleable 한 ERD 및 객체 설계						
	UX/UI 디자인						
개발	디자인 외주 및 확정						
	Github와 프로젝트의 연동을 통한 Dora metrics의 생산성 관련 지표 추출 서비스						
	개발자 개인 역량/개발 습관 지표 제공 서비스 개발						
	팀 당 위험 알림 서비스 제작 / Development cycle custom						
	feedback을 통한 추가 기획 수정 관련 개발						
테스트	1차 소프트웨어 마에스트로 CBT 및 feedback 수집						
	2차 공개 OBT 및 feedback 수집						
종료	개발 종료 및 서비스 배포						

○ 역할 분담

담당		역할 및 상세 활동
연수생	류동인	팀장, Infra Architecture 설계 및 구축, BE 개발
	백종현	FE 개발, BE 개발
	조인혁	BE 개발, DB 설계 및 구축
멘토	유저스틴	팀 개발 방향 리드 및 개발 방법론 조언, 아키텍처 조언
	이다니엘	팀 아이템 개발 지원 및 API 관련 조언, 기획 관련 조언
	최정현	팀 현장 리드 및 기획 관련 조언

○ 개발 시 예상되는 장애 요소 및 해결방안

- 실시간으로 Git의 정보를 가져오기 위하여 API 및 명령어를 적극적으로 사용해야 한다. 하지만 GitHub/BitBucket/GitLab 등의 개발 협업툴들은 다른 API를 사용하고 있어 고려할 부분이 존재한다. 개발초기부터 분석을 시작하고, 가능한 커버리지를 확실히 정해야 할 것이다.
- 저장 해야 할 데이터가 시계열 데이터이기 때문에, 데이터 수집 스케줄링 및 ERD 설계가 핵심일 듯 하다. 하지만, 확장성 있고 견고한 ERD를 개발 진행 이전에 설계가 가능할지에 문제가 예상된다. 이는 자유멘토링을 통해 해결할 계획에 있다.

□ 기대효과 및 활용 방안

○ 기대효과

사용자	개인과 팀의 성장을 막는 병목 지점을 식별하고, 이를 능동적으로 해결해 나갈 수 있다. 프로젝트 진행 상황에 대한 객관적인 지표를 확인하고, 이를 효과적으로 문서화 할 수 있다. 팀과 개인의 성장, 성과 측정을 자동화 해서 시간적, 비용적 이득을 얻을 수 있다.
비즈니스	수많은 팀의 데이터가 확보됨에 따라서 더욱 객관적인 생산성 평가 기준을 정할 수 있다. 데이터를 기반으로 팀의 문제점의 돌파구를 추천해 줄 수 있다. 팀이 직면한 문제의 해결책이 될 수 있는 웹솔루션을 추천함으로써, 광고 비용을 얻을 수 있다.
개발자	개발자들의 협업 과정과 개발 진행에 대해 이해하여, 개발 전반에 이해도를 높일 수 있다. 다양한 데이터를 모아 동적으로 보여주고 위험 감지하는 웹 프로젝트를 진행한다는 점에서, BackEnd와 FrontEnd의 이해도 향상. 커스텀마 이징 가능한 자율성 높은 툴을 개발하면서 다양한 디자인패턴을 적용해 볼 수 있을 것으로 기대됨

○ 결과물 활용방안

- 사업화 및 서비스 개선
 - 소프트웨어 마에스트로 과정 마지막에 진행하는 OBT를 통해 개발 시장 전반에서 사용할 수 있는 서비스로 개선
 - Variable한 팀 개발 환경 구성을 통해 학생 개발자 -> 현업 개발자까지 이어나갈 수 있는 사용자 경험 확충
- 시장 진입
 - 기업 인하우스 서비스 제공을 통한 전사적인 인사 관리 제공 예정
 - 추후 Infra 개선 및 서버 자원 확보를 통한 Public 서비스 제공 예정에 있음

□ 결과물 형태 및 서비스 방식

