SW마에스트로 제13기 프로젝트 기획서

**1. 프로젝트 개요서 *☞ 분량 : 1장으로 작성***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **프로젝트명** | # 프로젝트명 기재  DORA Matrix를 이용한 개발 생산성 향상 툴 | | | |
| **프로젝트 요약** | # 3줄 이내 제한  팀 단위에 개발에서 이루어지는 협업 과정에서의 진행 속도와 프로젝트의 안정성을, Google의 생산성 향상 관련 보고서를 기반으로 도식화하여 보여주어 좋은 개발 환경을 제공해주고, 진행에 문제가 있다면 개발자에게 정보를 제공한다. | | | |
| **기술 키워드** | # 머신러닝/딥러닝 등  JAVA Spring / React / Python | | | |
| **ICT연구개발**  **기술 분류** | ① (대분류) 웹 애플리케이션- ② (중분류) 생산성 툴, 애자일 | | | |
| **팀 명** | 알락꼬리여우원숭이 | **팀 원** | | 류동인, 조인혁, 백종현 |
| **목적 및 필요성** | 1. 팀 단위로 개발을 진행할 때, 팀의 성과를 높이는 방법이 필요하다.  2. 우리팀은 엘리트 팀인가? 팀원 각각은 그렇다고 생각하거나, 혹은 우리 팀의 개발 진행에 문제가 있다고 느낄 수 있다. 이를 객관적으로 평가받고 싶다.  3. 객관적으로 평가받은 후, 엘리트 팀에 가까워지고 싶다.  객관적 평가를 위해, 팀의 진행상황을 지표화, 수치화, 도식화를 통해 가시화 시키고, 생산성 관련 DORA metrics 기법을 이용하여 생산성을 향상시킨다. | | | |
| **프로젝트 개요** | 애자일 프로세스 기반의 팀 개발 진행에 있어 생산성 제고 및 지속 가능한 개발을 위한 방법론인 Dora Metrics를 이용한 프로젝트를 진행할 예정이다. | | | |
| 프로젝트 예상 기능   * Dora Matrix 지표 추출 및 제공(GitHub, BitBucket, GitLab)   + Coding ( Issue start -> Pull Request )   + PickUp ( Pull Request -> Review Start )   + Review ( Review Start -> Pull Request Merge )   + Deploy ( Merge -> Deployment ) * 개발 진행중 위험가능성이 있는 행동 감지 및 알림(Slack && Email)   + 평소보다 많은 PR -> 번 아웃 가능성   + Code Big Diff -> 한번의 PR에 너무 많은 코드 변경은 위험을 초래   + Too Many WIP(Working In Process) -> 너무 많은 WIP는 병목현상을 초래한다. * 개발자 개인의 코딩 습관 확인 ( commit frequency / error 후 revert 등) | | | |
| **수행 방법 및**  **추진 일정** | **6 : 서비스 기획, 아키텍쳐 설계, ERD 및 객체 설계, 디자인 확정**  **6~7 : Github와 프로젝트의 연동을 통하여 Dora metrics의 생산성 관련 지표 추출**  **8 : 개발자 개인의 역량/개발 습관 지표 제작.**  **8~9 : 서비스 개발 이후 소프트웨어 마에스트로 내부 CBT**  **9 : 팀 당 위험 알림 서비스 제작.**  **10~11 : CBT 결과 이후 유지보수 및 최종 공개** | | | |
| **기대 효과 및 활용 방안** | **리더** | 개발 전반의 문제점을 감지, 구성원들의 사기 및 문제점을 감지하여 팀 분위기를 더 건전하게 바꿀 수 있다. | | |
| **PM/스크럼 마스터** | 프로젝트의 진행 상황에 있어 걸림돌이 되는 요소를 식별 및 제거 할 수 있는 기회를 얻으며, 리더 및 팀 상황 상부 보고에 있어 객관적인 지표를 좀 더 효율적으로 얻을 수 있다. | | |
| **개발자** | 자신 및 팀원들의 상황을 파악, 개발 습관을 좀 더 건강하게 유지하여 더 나은 개발자로 성장할 수 있는 기회를 얻을 수 있다. | | |
| **팀 전체** | 개발 팀 전체의 효율성을 제고함으로써 더 높은 생산성을 기대할 수 있다. | | |
| **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 레벨 UI 노출 * **추가적으로,** Slack/Email을 통해 위험 요소 탐지 알림 * **개발자 개인의 개발 습관 파악 및 역량 인지를 위한 UI** | |

\* 작성 시 파란색 안내 문구는 삭제 후 내용 작성(본 문구 포함)

\* 프로젝트 개요서(1장으로 작성), 프로젝트 기획서 (5장 이내 작성), 발표 PPT(자유양식, 자유 분량) 제출

**2. 프로젝트 기획서 *☞ 분량 : 5장 이내 작성***

\* 항목 기재 요령

- 하단에 기재된 대항목(□)은 필수 기입 사항으로 추가, 삭제 등 변경 불가

- 대항목(□) 내 소항목(ㅇ)은 기존에 표기된 항목을 삭제하는 것은 불가하며, 필요에 따라 추가(ㅇ,-) 가능

- 주요 항목에 맞춰 자유롭게 작성

**□ 목적 및 필요성**

ㅇ 문제인식

* + 우리 팀은 엘리트 팀이 아니라는 판단을 하게 되었다.
  + 우리팀의 문제는 무엇일까? 라는 고민 하에, 우리는 우리 팀의 문제점을 객관적으로 판단할 수 없다는 결론을 내게 되었다.
  + 모든 문제는 문제를 파악한다는 것에서 시작되는 것인 만큼, 팀의 문제를 객관적으로 파악하고 수치화 및 도식화할 툴의 가능성을 찾기 시작했다.

ㅇ 기획 의도(문제해결)

* + 팀의 문제를 객관적으로 평가할 수 있는 검증된 지표가 필요했다.
  + Google Cloud의 DORA(DevOps Research and Assessment) 팀 보고서 2021 Accelerate State of DevOps Report에 따르면, 개발팀의 생산성을 평가할 수 있는 객관적인 지표(Deployment Frequency, Lead Time for Change, Mean Time to Recovery, Change Failure Rate)라는 요소가 있다고 한다. 그리고 이는 DORA Metrics라 불린다.
  + 이에 대해 해외에서는 DORA Metrics를 보여주고, 팀 생산성을 제고하기 위한 툴이 만들어졌다. 하지만 한국 개발 시장에 맞는 현지화 아이템은 나온 적이 없다.
  + 그렇기에 우리는 DORA Metrics를 한국 개발자 정서에 맞게 재해석하여 생산성 향상에 기여하고자 한다.

ㅇ 시장/소비자 동향/분석, 경쟁 제품 비교/차별화 요소

* + 주 타겟으로 하고 있는 한국 기업들의 경우, 2019년부터 관료식 경영 방식에서 애자일을 목표로 하고 있는 경우가 많아졌다.
  + 경쟁 제품들(LinearB, HayStack)
    - 공통
      * 오픈소스가 아닌 판매형 서비스, 개발자 당 달에 $20 정도
      * 상품성이 그 정도가 되진 않는다는 판단.
    - LinearB
      * 분석 없이 데이터 수집 후, 뿌려주는 것이 전부이다.
      * 직관적인 판단은 불가능한 상태로 뿌려 주기 때문에 직관적인 판단은 오히려 부족할 수 있음. 가독성이 떨어짐.
      * 팀 전반적인 위험은 알 수 없음. 위험 요소에 대한 판단은 없는 상황
    - HayStack
      * 솔루션 자체가 Dora Metrix 자체를 목적으로 한 것이 아닌, 상품 중 하나.
      * 그러한 이유로, 전반적인 툴 중의 하나로써 의 기능만을 담당함 그렇기 때문에 프로젝트 하나하나의 상황에 대해 알기에는 무리가 있다.
      * 알림의 요소가 개발팀의 문제를 파악하기에는 한계가 있다.(너무 적은 항목)
    - 차별점
      * 한국 개발자 시장에 맞는 언어 및 사용 경험 현지화.
        + 팀원 평가의 경우 개발자를 보호하기 위해 개발자에게만 노출시킨다.
        + 상위 관리자에게는 팀의 전반적인 상황과 Task별 업무 현황 파악을 용이하게 하기 위해 및 Alert 맞춤 설정가능.
      * 다양한 형태를 가진 팀들을 위한 솔루션
        + PR 또는 Deploy등을 안 하는 소규모 팀을 위해, 개발 Cycle Matrics factor를 선택 가능하게 한다.
        + GitHub 뿐만이 아닌, GitLab/BitBucket 등의 Private Repository를 사용하는 회사에서도 사용이 용이하도록 한다.
        + GitHub 또는 Organization의 프로필에 ReadMe.md에 팀의 랭크를 알 수 있는 API 제공.

**□ 프로젝트 개요**

ㅇ 프로젝트 소개

* + Google Cloud의 DORA(DevOps Research and Assessment) 팀 보고서 2021 Accelerate State of DevOps Report에 등장하는 생산성이 높은 팀의 git 사용 방식 및 개발 단계를 분석한 DORA Matrics를 이용, 건강한 Development Team 문화와 높은 생산성과 안정성을 도모.
  + 개발 속도에 집중하여 코드의 질을 떨어트릴 수 있는 LinearB, HayStack 등의 기존 Dora Matrics 이용한 상품들과 다르게, 개발 안정성을 추구하여 이식성 및 회복률이 높은 프로젝트를 만들 수 있도록 도모.
  + 개발팀의 리더 또는 DevOps가 수동으로 수집하여 파악해야 하는 업무 중 하나인 개발 Cycle 관리 및 개발 속도 관리를 자동화 함으로써, 더 혁신적인 아이디어 및 더 나은 팀을 위한 경영을 할 수 있도록 돕는 것이 최종 목표.

ㅇ 시스템 구성도

ㅇ 적용기술

|  |  |  |  |
| --- | --- | --- | --- |
| **구 분** | | **항 목** | **세부 내용** |
| S/W  개발환경 | OS | Ubuntu 18.04(SVR) | 다양한 개발 효용성을 위해 나온지 오래된 18.04(SVR) |
| 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) |

ㅇ 주요 기능

* + Dora Matrics에 기반, github에서 개발자들의 work를 추적. 팀 전반에 걸친 개발 cycle 및 working time 파악 기능.
  + 개발 중 Team에서 발생 할 수 있는 문제들 또는 눈 여겨 보아야할 이슈들(WIP, Too Short Review Time, Big Diff,
  + Burnout possibility 등)을 자동으로 파악, 알림을 보여줌으로써, 놓칠 수 있는 개발 이슈들을 파악할 수 있도록 하는 기능.
  + 서비스 사용 시점부터, 현재까지 개발팀에서 발생했던 문제와 개발 Cycle의 변화를 보여주며 더 나아갈 수 있는 방향성 제시.

ㅇ 개발 사양

* + 개발 및 테스트 서버 – AWS EC2, RDBS

**□ 수행 방법 및 추진 일정**

ㅇ 주요 기능별 수행방법

|  |  |
| --- | --- |
| **Team Git 정보 수집** | * Team Git accessKey 등록 후, git Organization 정보 검색 git script를 적용, 주기적인 정보 수집. * 더욱 상세한 정보를 파악하기 위해, selenium을 통한 크롤링 또한 |
| **Git을 통한 Dora Matrics 성취도** | * Redis 및 RDBS 저장 후, (JAVA Spring + React) Restful API 구조로 서비싱 |
| **개발 위험 알람** | * 주기적인 Team Git 수집 후, Slack Open API를 통해 알림 |

ㅇ 수행방법 확보방안(아웃소싱 등)

- 수원대 디지털 미디어 학과 졸업생 디자인 아웃소싱을 통한 UX/UI 디자인 개발.

- 팀원들의 UI 개발 능력이 부족하므로, 주기적인 FE(JS/React) 개발 스터디를 통해 FE 개발 능력 향상

- 서비스 환경 구축 이전, Docker Image를 통한 개발. 패키지 및 OS 버전 및 종류 차이로 인한 문제들을 최소화.

ㅇ 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 | 개발 및 테스트 서버 아키텍처 설계 및 세팅 |  |  |  |  |  |  |
| 서비스 서버 아키텍처 설계 |  |  |  |  |  |  |
| RDBS ERD 및 백엔드 객체 설계 |  |  |  |  |  |  |
| 개발 | **Team Git 정보 수집** |  |  |  |  |  |  |
| **Git을 통한 Dora Matrics 성취도** |  |  |  |  |  |  |
| **개발 위험 알람** |  |  |  |  |  |  |
| **CBT 결과 반영 및 코드 안정성 향상** |  |  |  |  |  |  |
| 테스트 | CBT 테스트 |  |  |  |  |  |  |
| 종료 | 개발 종료 및 서비스 배포 |  |  |  |  |  |  |

ㅇ 역할 분담

|  |  |  |
| --- | --- | --- |
| **담 당** | | **역할 및 상세 활동** |
| **연수생** | 류동인 | 팀장, Infra Architecture 설계 및 구축, BE 개발 |
| 백종현 | FE 개발 및 GitHub API 분석 및 적용 |
| 조인혁 | BE 개발 및 Slack API 분석 및 적용 |
| **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 |

ㅇ 개발 시 예상되는 장애 요소 및 해결방안

* 실시간으로 Git의 정보를 가져오기 위하여 API및 명령어를 적극적으로 사용해야 하는데, 이를 사용하기 위해서는 분석이 먼저 필수일 것 같음. 하지만, GitHub/BItBucket/GitLab 등의 개발 협업 툴들은 다 다른 API를 사용하고 있기 때문에 고민해봐야 할 부분이 있을 것 같다. 개발 초기부터 분석을 시작하고, 가능한 커버리지를 확실히 정해야 한다고 고려중.
* 개발을 진행함에 있어서, FE 역량이 부족한 팀이기 때문에 React를 사용함에 있어 불편함이 예상될 것 같음. 지속적인 스터디를 통해 해결해 나갈 생각.
* 저장 해야 할 데이터가 시계열 데이터이기 때문에, 데이터 수집 스케줄링 및 ERD 설계가 핵심일 듯 하다. 하지만, 관련하여 ERD를 Scaleable하게 작성해야 할 것인데 팀원들 모두가 의미 있는 ERD를 짜낼 수 있을 것에 대한 문제가 예상됨. 자유 멘토링을 통해 해결할 계획에 있음.

**□ 기대효과 및 활용 방안**

ㅇ 기대효과

|  |  |
| --- | --- |
| **사용자 측면** | 개발 전반의 문제점을 감지, 구성원들의 사기 및 문제점을 감지하여 팀 분위기를 더 건전하게 바꿀 수 있다. 또한 프로젝트의 진행 상황에 있어 걸림돌이 되는 요소를 식별 및 제거할 수 있는 기회를 얻으며, 리더 및 팀 상황 상부 보고에 있어 객관적인 지표를 좀 더 효율적으로 얻을 수 있다. 이러한 자동화를 통해 얻는 이점 또한 있겠지만 정확한 수치를 통해 더 나은 팀을 만들 수 있다는 장점이 있다. |
| **비즈니스 측면** | Agile 개발팀을 운영중인, 그리고 고려 중에 있는 많은 기업에 배포가 가능할 것으로 사료된다. 특히, 업무에 있어 자동화를 진행해야 하지만 비용 문제로 인해 고민중인 중간 크기 정도의 IT 개발팀 사용 업체를 대상으로 효과적일 것으로 판단하고 있다. |
| **개발자 측면** | 개발자 스스로의 성장에 있어서 가장 중요한 요소인 생산성과 안정성을 파악할 수 있다. 대부분의 환경에서, 개발자들의 역량 및 프로젝트 개발 적합도는 수치화 되지 않는다는 단점이 있다. 이 프로젝트는 이러한 면에서 개발자 스스로 자신의 역량을 파악하고, 어떠한 성향을 가진 개발자인지를 수치화 하여 보여줌으로써 더 큰 성장을 도모할 수 있다.  또한, 다양한 데이터를 쌓고 동적으로 보여주는 웹 프로젝트를 진행한다는 점에서, SystemOps/BE/FE를 지원하는 알락꼬리여우원숭이 팀원들에게도 좋은 프로젝트라고 할 수 있다. |

ㅇ 결과물 활용 방안

# 비즈니스 모델 구축, 사업화 계획(창업 등), 특허, 논문 등재, 기술 제휴 등

**□ 결과물 형태 및 서비스 방식**

ㅇ 결과물 형태

ㅇ 서비스 방식