개발팀 향후 계획 보고서

개발팀 류창선

1. 들어가기

개발팀에서는 앞으로 개선 및 보완할 점과 미작업 목록을 총 네 가지 키워드로 분류했습니다. 키워드는 각각 문서화, 관리, 개선, 성장이며, 구체적인 내용은 본론에서 설명드리도록 하겠습니다.

2. 본론

2.1. 문서화

2.1.1. 개발 프로세스 가이드라인

개발팀마다 다른 서비스를 개발하고, 그 개발 방식 역시 다릅니다. 또한 개발 절차나 협업하는 방법도 다릅니다. 그래서 개발 문화를 공유하는 지침서를 만들고자 합니다. 이것이 새로운 개발자의 적응에 도움을 줄 것이라 기대합니다.

2.1.2. 파트별 개발 산출물

예를 들어, API 명세는 프론트엔드 개발자와 백엔드 개발자 간의 소통 창구라 할 수 있으며, 아키텍처ᐧ레이어ᐧ컴포넌트 등 개발함에 있어서 미리 정의되어야 하는 부분이 있습니다. 이러한 부분들을 정리하여 개발 산출물로 만들고 지속적으로 유지 보수한다면 보다 탄탄한 서비스를 제공할 수 있을 것이라 생각합니다.

2.1.3. 저장소별 개발 세팅 가이드라인

현 개발팀에서는 다양한 개발 언어를 사용하고 있습니다. 그렇기 때문에 개발 진행 전 설정 단계에서 길을 헤매는 경우가 있습니다. 이러한 비효율을 해결하기 위해 설정 방법을 문서로 만들어 안내하려 합니다.

2.1.4. 버그 리포트

발생한 장애를 해결하는 것으로 그치지 않으려 합니다. 기록으로 남겨 원인을 파악하고, 환경 정보와 다양한 시도 방법, 그리고 해결 방법까지 담은 회사의 자산을 만들고, 공유하여 같은 장애가 발생하지 않게 조치하고자 합니다.

2.1.5. Git 사용법

형상 관리에 초점을 맞추려 합니다. 개발한 내역을 추적하여 장애를 미연에 방지하고, 문제가 생기는 부분은 뒤로 돌려 장애를 신속히 해결하고자 합니다. 이를 위해 형상 관리 시스템인 Git 사용법을 문서화하여 학습하겠습니다.

2.2. 관리

2.2.1. 서버 권한

현황 파악 결과, 1개만 존재하며, 탈취당하면 복구하기 힘들고, 모든 권한을 가지는 Root 계정만이 존재합니다. 그렇기 때문에 IAM 계정을 만들어 선택적으로 권한을 부여하여 서버 관리를 진행하고자 합니다.

2.2.2. 서비스 배포

운영 중인 서비스에 수정 사항을 반영하는 배포는 규칙이 필요합니다. 만약 규칙이 없다면 아무 때나 배포할 것이며, 이는 사용자에게 불편함을 줄 수 있을 것입니다. 일정한 검증 단계를 거쳐 정해진 요일에만 배포할 수 있는 정기 배포와 혹시 모를 비상 상황을 위한 긴급 배포 프로세스를 만들어 관리하겠습니다.

2.2.3. 서비스 버전

서비스 업데이트 사유는 다양합니다. 신규 기능이 추가될 수도 있고, 버그를 수정할 수도 있습니다. 수정 내역을 정리하여 릴리즈 노트라는 형태로 기록하면서 수정 내역마다 버전을 할당하는 방법으로 관리하고자 합니다.

2.2.4. Gitlab 개발/운영 통합

현재 개발팀에서는 서비스 소스 코드를 원격에 저장하는데, 그 저장 공간이 둘로 나눠져 있습니다. 이 둘을 하나로 합치는 작업을 통해 서비스 소스 코드를 효율적으로 관리하려 합니다.

2.2.5. GItlab CI/CD 구축

개발이 완료되면 크게 운영계와 개발계로 나눠 배포를 진행합니다. 그러나 현재 운영계와 개발계 모두 수동 배포를 하고 있으며, 절차 역시 둘(Gitlab - Jenkins)로 나눠져 있습니다. 절차를 하나(Gitlab)로 줄이고, 개발계만 자동 배포 가능한 환경을 구축하여 개발 리소스를 효율적으로 관리하겠습니다.

2.3. 개선

2.3.1. DB 데이터 암호화

개발팀에서 다루는 데이터 중 개인정보, 비밀번호 등 민감한 정보가 존재합니다. 이 데이터를 암호화하여 저장된 데이터를 보호하는 것이 이 개선 사항의 취지입니다. 구체적으로 펜타시큐리트 사의 D’Amo 솔루션을 도입해 적용하고자 합니다.

2.3.2. 어드민 조회 페이지 탭 분리

코스 어드민 서비스 가운데 조회 페이지는 총 네 개의 탭(대출실행 정보, 의뢰서 정보, 실행확인 조사, 권리 조사)으로 구성되어 있습니다. 일반적으로 탭마다 다른 데이터가 필요하므로, 서비스 소스 코드가 탭마다 분리되어 있어야 합니다. 하지만 현 조회 페이지는 하나의 서비스 소스 코드로 개발되어 있으며, 그 길이가 9천 줄이 넘습니다. 이로 인해 유지 보수 시 어려움이 있기에 탭 분리를 통해 서비스 로직을 개선하겠습니다.

2.3.3. Gitlab MR 템플릿 및 라벨 설정

서비스 소스 코드는 여럿이서 개발하고 있으므로, 서비스 소스 코드를 하나로 합치는 과정이 반드시 필요합니다. 이에 필요한 도구인 템플릿과 라벨을 모든 서비스 소스 코드 저장 공간에 추가하도록 하겠습니다.

2.4. 성장

2.4.1. 코드 리뷰

교학상장이라는 말이 있듯이, 개발자는 연차를 초월해 서로 배울 수 있습니다. 그래서 서로 배우기 위한 코드 리뷰 문화를 도입하고자 합니다. 최소 2인 이상의 교차 코드 리뷰로 자신이 개발한 소스 코드를 검증받고, 결함을 최소화하는 방향으로 나아가고자 합니다.

2.4.2. 개발 관련 스터디

개발팀 대다수가 3년 차 이하의 주니어 개발자입니다. 시니어 개발자들의 지식과 경험, 노하우를 공유하고, 함께 성장할 수 있는 다양한 주제의 스터디를 진행하겠습니다. 또한 개발팀뿐만 아니라, 회사 소속원이라면 차별 없이 스터디에 참여할 수 있게 전사 공지하는 방향도 고민하겠습니다.

2.4.3. RSS 피드 연동

기술 트렌드 학습과 기술 전문성 유지의 일환으로 네카라쿠배당토(네이버, 카카오, 라인, 쿠팡, 배달의민족, 당근마켓, 토스) 등 선진 기술 문화를 보유하고 있는 회사의 기술 블로그를 연동하는 작업을 시도해 보려 합니다. 각 기술 블로그에서 새로운 게시물이 등록되면 실시간으로 네이버웍스 메시지로 받아 볼 수 있게끔 조치해 보겠습니다.

3. 나가기

현재 개발팀은 운영 이슈 처리와 인수인계서 기반 로직 파악에 집중하고 있습니다. 신속히 안정화하여 향후 계획을 실행하겠습니다. 이상 개발팀의 향후 계획 보고를 마치겠습니다.