



권기영.

Project Manager

ttu445@gmail.com · 010-4953-9209

BUILT WITH AI

이 포트폴리오 웹사이트 역시 AI를 통해 직접 제작했습니다.

portfolio-giyeongkwon-nc.pages.dev

국회 정책비서, 게임 기획자를 거쳐 치열하게 라이브 서비스를 전개하는 환경 속에서 개발 PM으로 일하며, 제가 매일 마주한 현장은 늘 **물리적 한계와의 싸움**이었습니다. 대형 업데이트 시즌마다 폭증하는 번역량 걱정에 정작 인게임에 들어갈 실제 콘텐츠의 양이 적어지는 위험, 그리고 이로 인한 글로벌 반영 지연의 리스크는 늘 상존했습니다. 슬랙과 외부 캘린더를 바쁘게 횡단하는 수동 일정 조율과 파편화된 피드백의 휘발 역시 마찬가지였습니다. 이러한 기성 프로세스의 병목은, 현장의 페인포인트를 누구보다 잘 아는 사람이 직접 개선에 나서지 않으면 결코 해결할 수 없는 본질적인 갈증이었습니다.

이 고질적인 비효율의 관성을 깨기 위해, 저는 단순히 아이디어를 내는 PM에 머무는 대신, **직접 AI를 통해 시스템적으로 솔루션을 구현해 변화를 만드는 사람**으로 진화했습니다. 회의실 예약 봇부터 게임 전용 번역 시스템까지, 제가 구축한 모든 AI 솔루션은 단순한 기술적 실험이 아닙니다. 실무의 흐름과 아픔을 가장 깊이 이해하는 PM의 도메인 지식을 바탕으로 설계되어, 오퍼레이션의 임계치를 정면 돌파하고 조직의 실질적인 가치를 증명해 낸 강력한 '제품'들입니다.

"안 된다"는 한계 대신 기술로 **"되는 구조"를 설계하는 PM**으로서, 업무 프로세스를 분석·재설계하고 작은 PoC로 기술 타당성을 검증해 조직의 공식 프로세스로 확산시키는 흐름을 잘 다룹니다. 이제 정책비서 시절 다져진 이해관계자 조율 감각과 5팀 매니징·2주 릴리즈 사이클·외주 운영에서 쌓은 게임 개발 프로세스의 실행 디테일을 함께 들고, **엔씨소프트 AX지원실의 전사 AI 전환**이 '도입된 기술'이 아니라 '바뀐 일하는 방식'으로 남도록 만드는 사람이 되겠습니다.

INCLUDED PROJECTS

- 01 이터널 리턴 전용 AI 번역 시스템
- 02 Slack 기반 업무 완결형 AI 회의실 봇
- 03 성우 녹음 일정 자동 등록 봇
- 04 이터널 리턴 보이스 제작 워크플로우 통합 시스템
- 05 Jira 스웬레인 자동 생성 봇
- 06 콜라보 종합상황판
- 07 슬랙 점심 주문 요약 봇
- 08 라이브 오퍼레이션 및 글로벌 프로젝트 매니징
- 09 팀 관리 및 정무적 리더십

이터널 리턴 전용 AI 번역 시스템.

우리 게임의 컨텍스트를 이해하는 전용 시스템으로 번역 파이프라인을 혁신하다

ROLE	PERIOD	TYPE
기획설계 · 도입 추진 · 운영 설계 · 시스템 개선	2026.01 – Present	사내 시스템 도입 및 운영 프로세스 혁신

01 · PROBLEM

업무 폭증과 대응 속도의 물리적 한계

대형 콜라보레이션이나 시즌 업데이트 시기에는 번역 업무량이 평시의 임계치를 상회하게 됩니다. 이 과정에서 다음과 같은 잠재적 리스크와 실질적인 페인 포인트가 발생할 수 있습니다.

물리적 대응의 한계 — 업무량이 인게임 반영 지연을 초래할 수 있을 만큼 방대해지면서, 실무자들이 매번 야근을 불사하며 일정을 맞추는 상황이 반복되었습니다.

글로벌 대응 속도 저하 — 팀에서 최선을 다해 대응했음에도 불구하고, 긴박하게 돌아가는 라이브 서비스 특성상 글로벌 공지 등이 유저 기대치보다 늦게 나가는 경우가 발생하며 속도감에 아쉬움이 남기도 했습니다.

시중의 범용 LLM이 가지는 한계

범용 AI는 게임 도메인 고유의 '약속된 언어'를 알지 못합니다. 예를 들어, 캐릭터 스킬명인 "최후의 발악"을 단순 LLM에 넘기면 문자 그대로 "Last Struggle"로 직역합니다. 하지만 우리 게임의 컨텍스트를 반영한 의도된 번역은 "Last-ditch Effort"여야 합니다. 이러한 고유 명사와 뉘앙스의 오역이 빌드 전체에 누적되면 글로벌 서비스의 몰입도가 근본적으로 흔들리게 됩니다.

02 · APPROACH

'이터널 리턴'만을 위한 전용 로컬라이제이션 시스템

단순한 텍스트 변환기를 넘어, 우리 게임에 완벽하게 최적화된 다층 구조의 전용 시스템을 설계하는 것을 목표로 삼았습니다.

전용 엔진 설계 — 범용 LLM의 자연스러움 위에 16만 개 이상의 사내 마스터 용어집을 '강제 치환 레이어'로 얹어, 우리 게임만의 고유한 톤앤매너를 유지하도록 했습니다.

하이브리드 워크플로우 — 번역가가 '기준'을 세우고 AI가 '반복'을 해결하는 구조를 통해, 사람이 물리적 한계에 부딪히지 않고도 품질을 유지할 수 있는 환경을 구축했습니다.

사용자 피드백 루프 운영 — 현장에서 발견된 문제를 빠르게 시스템 개선으로 연결하기 위해, 전용 피드백 채널을 개설해 운영했습니다.

🔒 rnd_ai번역피드백

고객님이 이 채널을 생성한 날짜는 날짜: 2025년 12월 18일입니다. 🔒 rnd_ai번역피드백 채널의 맨 첫 부분입니다.

👤 채널에 사람 추가 📄 설명 추가 ✉️ 채널에 이메일 보내기 🗑️ 템플릿 선택

2025년 12월 18일 ▾

권기영 오전 11:12
rnd_ai번역피드백에 참여했습니다. 또한 초대를 통해 [redacted] 명의 다른 사용자가 참여했습니다.

권기영 오전 11:14
안녕하세요 구두로 말씀드렸던 ai번역 관련 피드백을 위한 채널입니다. 낱말의 피드백도 좋으니 사용해보시고 많은 피드백 부탁드립니다.

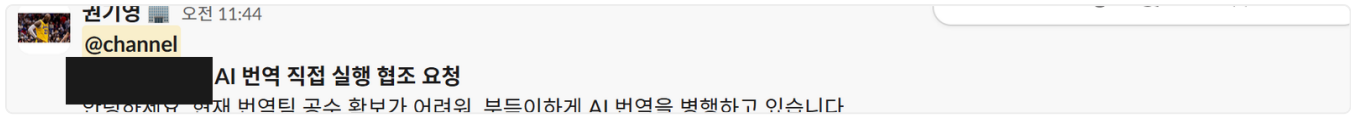
👍 3 🗨️ 1 🔄

04 · IMPACT

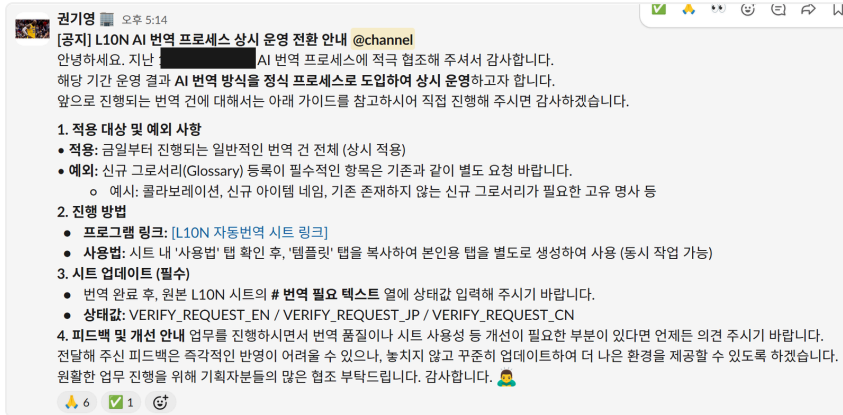
현장의 문제를 해결하고 공식 프로세스가 되다

효율과 퀄리티가 현장에서 즉각적으로 증명되며, 본 시스템은 이터널 리턴의 정규 워크플로우로 완전히 자리 잡았습니다.

처음에는 번역팀 캐파를 보완하기 위한 **임시 협조 요청**으로 출발했지만,



이후 효율과 퀄리티가 검증되며 **상시 운영 프로세스로 정식 전환**됐습니다.



05 · LEARNINGS

PM의 도메인 지식이 곧 AI의 품질이다

이 프로젝트는 “기술보다 중요한 것은 기술을 어디에, 어떻게 붙일 것인가를 아는 안목”이라는 것을 증명한 사례였습니다.

번역팀과 보이스 녹음 업무 전반을 매니징하며 수많은 자료가 외주 업체와 오가는 복잡한 파이프라인을 직접 관리했습니다. 타이트한 스케줄 속에서 발생하는 병목 구간을 피부로 느꼈고, 수만 건의 데이터를 검토하는 과정에서 우리 게임만의 고유한 컨텍스트와 도메인 지식을 깊게 쌓을 수 있었습니다.

이러한 현장의 경험이 있었기에 16만 개의 용어집을 시스템에 촘촘히 엮어낼 수 있었고, 결과적으로 AI가 단순한 실험적 툴을 넘어 실제 조직의 임팩트로 작동하게 만들었습니다. PM이 실무의 흐름과 아픔을 깊이 이해하고 있을 때, AI 기술은 비로소 조직의 진정한 해결책이 된다는 확신을 얻었습니다.

Slack 기반 업무 완결형 AI 회의실 봇.

조율부터 예약까지, Slack을 벗어나지 않는 업무 완결형 Agent

ROLE	PERIOD	TYPE
기획 · 개발	2026.03 – Present	사내 커뮤니케이션 흐름에 결합된 업무 자동화

01 · PROBLEM

조율과 예약 사이에서 끊기는 워크플로우

님블뉴런은 NAVER Works Calendar를 통해 회의실을 예약합니다. 하지만 실제 회의 조율은 대부분 Slack 채널에서 이루어집니다. 특히 개발리뷰 시즌에 많은 회의로 일정이 바쁜 개발 리더들이 포함된 회의의 경우, [리더들의 빈 시간 확인 → 회의실 가용 상태 대조 → 참석자 전원 일정 확인] 과정을 수동으로 반복해야 했습니다. 슬랙 스레드에서 논의하다가 예약을 위해 외부 툴로 이동하는 과정 자체가 맥락을 끊는 업무 비용이자 병목이었습니다.

02 · APPROACH

Slack 안에서 업무가 종결되는 Agent 구조

단순히 질문에 답하는 챗봇을 넘어, 논의가 일어나는 Slack 안에서 즉시 업무가 종결되는 Agent 구조를 설계했습니다.

사용자 중심의 인터페이스 — 새로운 툴에 적응할 필요 없이, 사용자가 가장 익숙한 Slack에서 평소 대화하듯 자연어로 요청하는 방식을 채택했습니다. “이번 주 금요일 오후에 아티렉터, 프로그래머1, 기획자1님 모두 가능한 시간에 회의실 어디야?”라는 한 문장이면 충분합니다.

지능형 교차 분석 — AI가 요청을 해석해 특정 참석자들의 개인 캘린더와 회의실 현황을 실시간으로 교차 분석하고, 모두가 만족하는 최적의 옵션을 도출합니다.

시스템 자동화의 완결 — 슬랙에서의 대화로만 끝나지 않고, 최종 확정 시 NAVER Works Calendar에 실제 일정 등록까지 자동으로 수행합니다. 이를 통해 ‘커뮤니케이션’과 ‘행정 입력’ 사이의 간극을 완전히 제거하는 것을 목표로 했습니다.

03 · BUILD

실제 문제를 해결하는 핵심 기능

- 자연어 기반 컨텍스트 파싱 — 슬랙 스레드 내 날짜, 시간, 참석자 정보 추출
- 교차 일정 최적화 — 특정 참석자(개발 리더 등)의 빈 시간과 회의실 가용 상태를 실시간 대조
- 인터랙티브 UI 제공 — 분석된 가용 시간대 중 사용자가 선택할 수 있는 슬랙 UI 버튼 제공
- API 통합 및 정합성 유지 — NAVER Works API를 통한 일정 등록 및 외부 수정 사항 실시간 반영
- 연속성 있는 대화 처리 — 슬랙 스레드 문맥을 유지하여 “그때 말고 한 시간 뒤로 변경해줘” 같은 후속 요청 처리

권기영 3월 19일 오전 11:00
회의실봇 안내

회의실 예약/조회를 도와주는 봇입니다.
필요한 채널에서 회의실 봇을 초대 후 @회의실봇 멘션하거나 @회의실봇 DM으로 자연어 질문하면 됩니다.

사용 예시

- 오늘 회의실 현황 → 시간대별 예약되어있는 회의실 확인
- 내일 오후 2시 빈 회의실 → 빈 회의실 목록 + 예약 버튼 표시
- 오늘 홍길동 김철수 이영희 전부 가능한 시간 → 여러 사람 공통 가능 시간 확인
- 이번주 목요일 전체 예약 현황 → 누가 어디 예약했는지 확인

예약 방법
빈 회의실 조회하면 각 회의실별 예약 버튼이 나옵니다. 버튼 누르고 회의 제목/참석자만 입력하면 네이버 워크플레이스 캘린더에 자동 등록됩니다.

시간/날짜는 “오늘”, “내일”, “이번주 금요일”, “오후 3시” 같이 자연어로 말하면 알아서 인식합니다.
혹시나 작동 상에 문제가 생기거나, 혹은 개선 피드백 혹은 문의사항 있으면 @권기영 멘션 주세요!

04 · IMPACT

조율 단계가 사라지고 실행만 남다

여러 명의 캘린더를 일일이 대조하며 회의실을 찾던 번거로움을 없앴습니다. 사용자는 슬랙을 벗어나지 않고도 '가장 합리적인 예약 가능 시간'을 추천받음으로써, 의사결정에서 실행(예약)까지의 리드타임을 획기적으로 줄였습니다. AI를 단순한 정보 제공자가 아닌, 복잡한 운영 예외(충돌/수정)를 처리하는 효율적인 **운영 보조 인력**으로 포지셔닝한 사례입니다.

05 · LEARNINGS

AI의 품질은 '모델'보다 '문제 정의와 예외 설계'에서 온다

이번 프로젝트를 통해 생성형 AI의 가치는 화려한 모델 성능보다, 실제 업무 현장의 복잡한 예외 처리(데이터 정합성, 일정 충돌 로직 등)를 얼마나 꼼꼼하게 설계하느냐에 달려 있다는 것을 체감했습니다. 이후 자동화 도구를 기획할 때도 '어떤 모델을 쓸까'보다 '사용자의 흐름 속에서 어떤 예외가 발생할까'를 먼저 고민하는 습관을 얻게 되었습니다.

성우 녹음 일정 자동 등록 봇.

Slack 확정이 곧 Google Calendar 등록과 리마인드로 이어지는 외주 협업 자동화

ROLE

기획 · 개발

PERIOD

2026.03 – Present

TYPE

외주 스튜디오와의 협업 비효율 개선

01 · PROBLEM

대화 속에 흘러가 버리는 일정과 행정 부채

성우 녹음 일정은 외부 스튜디오와 가안 조율 후 Slack에서 사내 실무자들의 최종 확인을 받아 확정됩니다. 문제는 확정 이후, 제가 직접 Google Calendar를 켜서 수동으로 입력해야 외주 스튜디오 PM과 싱크가 맞는다는 점이었습니다. 여기에 대본 전달 같은 내용도 일정에 따라 직접 챙겨야 했기에, 이러한 행정 업무의 연쇄는 늘 심리적 부채이자 업무 흐름을 끊는 병목이었습니다.

02 · APPROACH

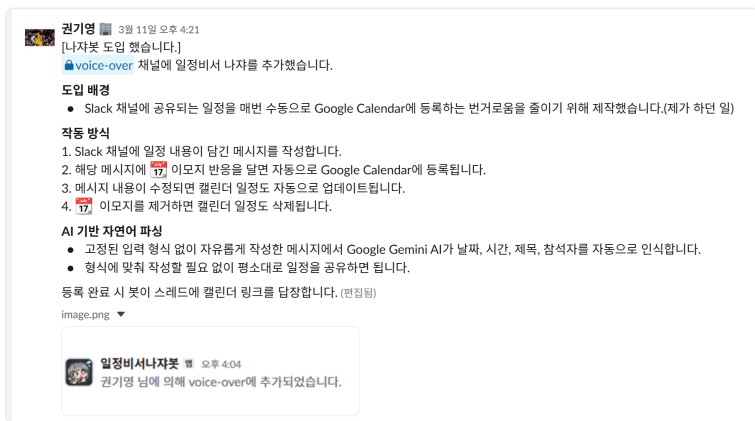
이모지 반응만으로 등록되는 마찰 없는 흐름

제가 수행하던 반복 업무를 시스템에 이관해, “확정 선언”이 곧 “데이터 등록”이 되는 직관적인 흐름을 설계했습니다. 슬랙 메시지에 이모지 반응 하나만 달면 날짜·캐릭터·시간 정보를 파싱해 Google Calendar에 자동 등록합니다. 또한, 일정에 맞춰 대본 등 필수 자료 전달을 개인 DM으로 재촉하는 리마인드 시스템을 구축하되, 버튼 토크식으로 보낸 자료는 다시 리마인드 오지 않게 하여 알림 피로도를 낮췄습니다.

03 · BUILD

실제 문제를 해결하는 핵심 기능

- Slack 메시지 이모지 반응 기반의 등록 트리거
- 자유형 텍스트에서 일정 데이터 파싱 — 성우·시간·캐릭터 등
- 외주 스튜디오 공유용 Google Calendar 등록·삭제 동기화
- D-Day 기반 리마인드 DM — 자료별 버튼 토크 체크리스트



04 · IMPACT

비정형 텍스트가 구조화 데이터로, 그 위에 후속 액션이 흐르다

슬랙에서의 “내부 확정”이 별도 작업 없이 “대외 공지”와 “사후 관리”로 직결되는 체계를 만들었습니다. 캘린더 복사-붙여넣기 같은 단순 반복 행정을 제거하고, 시스템이 마감 기한을 먼저 챙겨주어 에셋 누락 리스크를 차단했습니다. 도구 간 이동에 따르는 인지적 비용을 없애고, 본질적인 업무에 더 집중할 수 있는 환경을 스스로 구축했습니다.

05 · LEARNINGS

마찰 줄이기는 입력만이 아니라 후속 흐름까지의 문제

입력의 마찰만 줄여서는 사용자가 도구 안에 머물지 않는다는 걸 확인했습니다. 등록 이후의 리마인드와 체크리스트까지 같은 흐름 안에서 처리되어야 비로소 자동화의 가치가 완성됩니다. 서로 다른 도구를 사용하는 조직 간의 '브릿지 역할'을 자동화함으로써, 외부 조직과 제 개인의 리드타임을 함께 단축할 수 있었습니다.

이터널 리턴 보이스 제작 워크플로우 통합 시스템.

흩어진 4개의 도구를 하나의 흐름으로 — PM이 직접 설계하고 개발한 보이스 제작 파이프라인

ROLE	PERIOD	TYPE
기획 설계 · 직접 개발 (AI 에이전트 활용) · 운영	2026.05 – Present	사내 워크플로우 시스템 1인 자체 개발

01 · PROBLEM

PM이 '인간 라우터'가 되어버린 8단계 파이프라인

게임 캐릭터 보이스 제작은 번역팀, 사내 실무자, 외부 녹음 스튜디오가 얽힌 8단계 파이프라인입니다. 문제는 이 8단계가 Google Sheets, Gmail, Drive, Slack — 서로 연결되지 않는 4개의 도구에 흩어져 있었고, 그 사이를 잇는 유일한 연결고리가 PM의 수작업이었다는 점입니다.

모든 단계가 수동 라우팅 — 일정은 시트에 색칠하고, 발주는 Gmail에 매번 새로 작성하고, 녹음본이 도착하면 Drive 링크를 Slack에 복사해 나르고, 실무자들이 Slack에 남긴 셀렉 의견을 PM이 다시 메일로 옮겨 스튜디오에 전달했습니다. 콘텐츠 하나가 끝날 때까지 같은 정보가 사람 손으로 네 번씩 이사를 다녔습니다.

흩어진 커뮤니케이션, 불가능한 인수인계 — 어떤 콘텐츠가 지금 어느 단계인지, 스튜디오와 무슨 합의가 오갔는지가 메일함과 슬랙 스레드와 시트에 파편화되어 있어, 담당자가 아니면 현황 파악 자체가 불가능했습니다. 업무를 넘겨받을 다음 사람을 위한 구조가 존재하지 않았습다.

시트 하나의 겸직 한계 — 스프레드시트 하나가 일정 트래커, 캐릭터 설정집, 성우 캐스팅 현황을 동시에 떠맡으며 구조적 한계에 도달해 있었습니다.

02 · APPROACH

도구를 하나 더 얻는 대신, 흐름 자체를 다시 설계하다

출발점은 앞서 만든 성우 녹음 일정 자동 등록 봇이었습니다. 8단계 중 '일정 등록' 한 구간을 자동화해 실제로 운영해 보니, 마찰이 사라진 구간 옆에서 나머지 구간들의 수동 라우팅이 오히려 더 선명하게 보였습니다. 구간마다 봇을 하나씩 덧붙이는 방식으로는 결국 '봇들 사이를 잇는 일'이 또 PM의 몫으로 남는다는 것도요. 그래서 아이디어를 더 큰 맥락으로 확장했습니다 — 발주부터 셀렉 파일 정리까지, 흐름 전체를 한 곳에서 컨트롤하는 단일 시스템. 또 하나의 관리 도구를 추가하는 대신 흐름 자체를 다시 설계하기로 하고, 외주 개발 없이 직접 만들기로 했습니다.

스펙 주도 개발 — PM이 직접 만든다 — AI 코딩 에이전트를 페어 프로그래머로 두고, 기능마다 [브레인스토밍 → 설계 스펙 → 단계별 구현 플랜 → 구현 → 2단계 코드 리뷰]의 프로세스를 돌렸습니다. 실작업일 10일 동안 설계 스펙과 구현 플랜 문서 20여 건이 쌓였고, 모든 기능이 문서에서 출발해 코드로 끝났습니다.

단계별 출시 — 매 스테이지가 실무 투입 가능 — 전체를 5개 스테이지로 쪼개, 시트 대체(일정·캐릭터·캐스팅)부터 출시하고 Gmail 자동화, Drive-Slack 통합, 셀렉 워크플로우를 순차적으로 없었습니다. 완성을 기다리지 않고 현장에 먼저 투입하는 구조입니다.

3-role 권한 모델 — PM(전체), 실무자(셀렉·열람), 스튜디오(캐스팅 정보만 보이는 락다운 화면)로 권한을 분리해, 외부 협력사까지 한 시스템 안으로 끌어들이었습니다.

03 · SYSTEM

8단계 워크플로우가 앱 안에서 흐른다

단계	기존	시스템에서
러프 일정	시트 색칠	Gantt 스케줄 보드
스튜디오 발주	Gmail 수동 작성	템플릿 + 변수 자동 치환 → 원클릭 발사
일정·자료 협의	메일함에 파묻힘	쓰레드 뷰어 내장, 콘텐츠 카드에서 바로 답장
녹음본 공유	Drive→Slack 수동 복붙	파일 등록 + 슬랙 알림 (멘션 자동완성)
셀렉	Slack 의견 → PM이 메일로 중계	실무자가 앱에서 직접 듣고 분류
셀렉 파일 정리	Drive 수동 이동	"셀렉 완료" 클릭 → 폴더 자동 생성 + 일괄 복사 + 슬랙 알림

핵심은 셀렉 워크플로우입니다. 실무자가 셀렉 페이지를 열면 편집본 폴더의 음원이 자동으로 나열되고, 브라우저에서 바로 재생하며 셀렉본 / 업적 / 제외로 분류합니다. 진행 상태는 저장되어 언제든지 이어할 수 있고, "셀렉 완료"를 누르는 순간 분류 결과대로 Drive에 폴더가 만들어지고 파일이 복사되며 슬랙으로 완료 알림이 나갑니다. 기존에 Slack 의견 취합 → PM 메일 중계 → Drive 수동 이동으로 이어지던 채널 라우팅 구간이 통째로 사라졌습니다.

기술적으로는 Next.js 15 + Supabase 기반의 풀스택 앱으로, Drive 음원을 서버 프록시로 스트리밍(HTTP Range 지원)해 토큰 노출 없이 재생하고, Slack 워크스페이스 멤버를 동기화해 @ 입력만으로 실제 핑이 가는 멘션을 완성하는 등 현장 사용감을 좌우하는 디테일에 공을 들였습니다. Cloudflare Workers에 배포해 운영 중입니다.

04 · IMPACT

10일, 1인, 실서비스

PM 1인이 실작업일 10일 만에 실서비스 배포까지 — 커밋 190+, TypeScript 약 12,000줄, DB 마이그레이션 22개. 별도 개발 리소스 요청 없이 기획자가 직접 설계하고 구현해 사내 배포까지 완료했습니다.

수동 라우팅의 구조적 제거 — 발주 메일 작성, 녹음본 링크 나르기, 셀렉 의견 중계, Drive 파일 정리 — PM의 손을 거치던 반복 구간이 시스템의 버튼 하나로 대체되었습니다.

인수인계가 가능한 업무로 — 콘텐츠별 진행 단계, 메일 쓰레드, 파일 위치가 한 화면에 모여면서, '담당자의 머릿속에만 있던 업무'가 누구나 이어받을 수 있는 구조로 바뀌었습니다.

05 · LEARNINGS

현장을 아는 사람이 직접 만들 때 생기는 일

AI 번역 시스템 프로젝트에서 "PM의 도메인 지식이 곧 AI의 품질"이라는 확신을 얻었고, 그 후로 일정 봇·Jira 자동화 같은 도구들을 직접 만들어 현장에 정착시켜 왔습니다. 이 프로젝트는 거기서 규모를 한 단계 더 밀어붙인 검증이었습니다 — 단일 기능 봇을 넘어, 권한 모델과 DB, 외부 협력사까지 품는 풀스택 운영 시스템까지도 PM이 직접 만들 수 있는가.

보이스 제작 파이프라인을 수년간 직접 굴리며 어느 단계에서 정보가 유실되고 누구의 야근으로 메워지는지를 피부로 알고 있었습니다. 그래서 요구사항을 누군가에게 '설명'할 필요 없이, 페인 포인트에서 곧바로 화면과 데이터 모델을 설계할 수 있었습니다. 외주가 압축 파일로 줄지 폴더로 줄지, 실무자가 셀렉을 끊었다 이어할지 같은 — 현장에 있지 않으면 스펙에 적을 수 없는 디테일들이 시스템의 완성도를 결정했습니다.

AI 코딩 에이전트는 그 설계를 코드로 바꾸는 속도를 극적으로 끌어올렸지만, 무엇을 만들지·어디까지 만들지·무엇을 만들지 않을지의 판단은 끝까지 사람의 몫이었습니다. 기획과 개발의 경계가 흐려지는 시대에, PM의 경쟁력은 '현장의 아픔을 시스템의 언어로 번역하는 능력'이라는 것을 몸으로 확인한 프로젝트입니다.

Jira 스웸레인 자동 생성 봇.

새 버전 발행을 감지해 Jira Kanban 보드에 스웸레인을 자동 생성하는 운영 자동화

ROLE

기획 · 개발

PERIOD

2026.03

TYPE

Jira 운영 반복 업무 자동화

01 · PROBLEM

버전이 생길 때마다 반복되던 보드 운영의 수작업 병목

라이브 서비스 운영에서는 새 버전이 생길 때마다 Jira Kanban 보드에 해당 버전용 스웸레인을 추가해야 했습니다. 이 작업은 단순해 보이지만, 여러 보드에 반복 적용해야 하고 누락 시 실무자들이 새 버전 기준으로 업무를 정렬하지 못하는 문제가 있었습니다. 특히 버전 추가는 주기적으로 발생하는데, 매번 Jira 보드 설정에 직접 들어가 이름을 만들고 JQL을 넣고 순서를 맞추는 과정은 운영 집중도를 떨어뜨리는 소모적 관리 업무였습니다.

02 · APPROACH

"새 버전 생성"이 곧 "보드 운영 반영"으로 이어지도록 설계

반복되는 보드 관리 작업을 사람의 기억이나 수동 대응에 맡기지 않고, 새 버전의 생성 자체를 트리거로 삼아 자동 반영되도록 만들었습니다. Jira의 버전 목록을 주기적으로 감시해 새 버전이 확인되면, 지정된 여러 Kanban 보드에 해당 버전용 스웸레인을 일괄 생성하는 흐름을 설계했습니다. 단순 생성에 그치지 않고, 이미 존재하는 스웸레인과 비교해 더 높은 버전만 추가되도록 하여 중복 생성이나 과거 버전 오염도 막았습니다.

03 · BUILD

실제 운영에 맞춘 핵심 기능 구현

- Jira 버전 목록 주기 감시 — 새 버전 자동 탐지
- 여러 보드 대상 스웸레인 일괄 생성
- 기존 스웸레인 조회 후 최신 버전보다 높은 경우에만 추가
- Puppeteer 기반 Jira UI 자동화 및 로그인 세션 유지
- 수동 실행 (`node index.js 12.0.0`) + 자동 감시 (`npm run watch`) 이중 지원

04 · IMPACT

버전 관리가 개인 작업이 아니라 시스템 반응이 되다

새 버전이 나올 때마다 사람이 Jira 보드 설정을 반복적으로 손보던 흐름을 제거했습니다. 그 결과 보드 운영 반영이 더 빨라지고, 여러 보드 간 적용 누락 가능성도 줄었습니다. 특히 업무물의 관리성 문제를 사람의 꼼꼼함으로 해결하는 대신, 시스템이 먼저 반응하도록 바꾸면서 버전 단위 업무 정렬의 일관성을 확보했습니다. 작지만 자주 발생하는 관리 업무를 줄여 팀의 운영 리듬을 더 매끄럽게 만들었습니다.

05 · LEARNINGS

운영 자동화의 핵심은 기능 구현보다 '누락 가능성 제거'에 있다

이 작업을 통해 운영 자동화는 시간을 몇 분 아끼는 문제를 넘어서, 반복 업무에서 발생하는 누락과 편차를 구조적으로 제거하는 일이라는 점을 확인했습니다. 또한 Jira처럼 완전한 공개 API만으로 해결되지 않는 도구는 UI 자동화와 내부 API 활용을 함께 고려해야 실무적인 해법이 된다는 것도 배웠습니다. 결국 중요한 것은 거창한 시스템이 아니라, 팀이 반복적으로 겪는 작은 불편을 정확히 제거하는 자동화였습니다.

콜라보 종합상황판.

시트·슬랙·Jira·드라이브 사이의 공백을 메우는 협업 허브 — 산재된 정보가 단일 화면으로 응집되는 게임 콜라보 운영 도구

ROLE	PERIOD	TYPE
기획 · 개발	2026.05 – Present	콜라보 포스트모템 액션 아이템 실행 — 외주·내부·IP 홀더 협업 비효율 개선

01 · PROBLEM

정립된 프로세스와 실제 현장 사이의 깊은 괴리

이전 콜라보 운영 회고에서 가장 두드러진 부채는 기존 프로세스와 실제 업무 흐름 사이의 깊은 괴리였습니다. 명문화된 절차는 있었지만 실제 작업물은 개인 로컬에, 피드백은 슬랙에, 일감은 Jira에 파편화되어 있었습니다. “최신 버전이 어디인가”라는 단순한 질문에도 도구 4개를 횡단해야 했고, 그 사이 IP 홀더의 핵심 피드백은 빠른 대화 흐름 속으로 휘발되곤 했습니다. 프로세스가 실제 업무 속도를 따라가지 못해 정보가 걸도는 상황이었습니다.

02 · APPROACH

한 화면에서 모든 컨텍스트가 종합되도록

도구를 늘리는 대신 도구 간의 다리를 놓아 괴리를 좁히는 방향을 택했습니다. ‘한 콜라보 = 한 화면’이라는 원칙 아래, 마스터 폴더(드라이브), 일감 필터(Jira JQL), 추적 채널(슬랙)을 단일 뷰로 응집시켰습니다. 리소스 행 옆에는 피드백, 슬랙 대화, Jira 하위 티켓을 한 클릭으로 호출하는 사이드 패널을 두어 컨텍스트 스위칭을 최소화했습니다.

03 · BUILD

실제 문제를 해결하는 핵심 기능

- 콜라보별 통합 셋업 — 탭마다 드라이브 URL / 슬랙 채널 / Jira JQL이 프리셋으로 묶여 자동 트래킹
- 실시간 협업 동기화 — Supabase Realtime을 활용해 동료의 변경 사항을 새로고침 없이 즉시 반영
- Jira 양방향 지능형 동기화 — 5분 폴링으로 상태·담당자 등을 동기화하되, 사용자가 편집 중인 항목은 충돌 방지를 위해 자동 스킵
- Jira 계층형 다중 매칭 — Epic-Task-Subtask 구조를 인식해 하나의 리소스를 여러 담당자(그래픽·사운드 등)와 연결하고 진행률을 자동 종합
- 슬랙 AI 자동 분류 — CF Worker와 Gemini가 30분마다 대화를 분석해 관련 리소스에 매칭. 오분류는 휴지통-복원 흐름으로 간편하게 정정
- IP 홀더 피드백 패널 — 피드백 발생-반영-최종 확인 단계를 체크박스로 추적. 듀얼 모니터 워크플로우를 고려해 별도 창 분리 지원
- 통합 대시보드 — 진행률, QA 검수 현황, 마감 임박 건, 미해결 피드백을 한눈에 조망

04 · IMPACT

흩어진 정보가 하나의 흐름으로 묶이다

도구 이동에 따르는 인지 비용이 제거되어, 운영자가 화면을 벗어나지 않고도 콜라보의 전체 맥락을 파악할 수 있게 되었습니다. 슬랙에서 휘발되던 IP 홀더 피드백은 영구 기록되어 누락이 사라졌고, AI가 관련 대화를 자동으로 행 옆에 모아주어 검색 비용을 획기적으로 낮췄습니다. 포스트모템이 지적한 4대 액션 아이템(마스터시트 / 드라이브 / 히스토리 / 피드백 트래킹)을 단일 도구로 해결하여, 차기 콜라보에서 동일한 부채가 반복되지 않는 기반을 마련했습니다.

05 · LEARNINGS

도구는 데이터가 아니라 사람의 흐름을 따라야 한다

이번 작업은 "새 도구를 만든다"보다 "사람이 이미 일하는 방식 위에 데이터를 흐르게 한다"가 훨씬 중요하다는 점을 다시 확인시켜 줬습니다. 운영자는 한 콜라보를 운영할 뿐, 슬랙·드라이브·Jira·시트로 나누어 사고하지 않습니다. 그래서 도구를 더 추가하기보다 도구 사이의 다리를 놓아, 한 화면에서 컨텍스트가 자연스럽게 종합되도록 만드는 방향이 옳았습니다.

또한 AI 자동화는 **통제 가능성**이 핵심이라는 점도 체감했습니다. 모든 걸 한 번에 끝내는 일괄 처리보다, "**사용자 트리거 + 결과 캐시**" 흐름이 운영 안정성 면에서 훨씬 오래 살아남습니다. 오분류가 발생하면 사용자가 휴지통-복원으로 즉시 정정할 수 있는 안전장치를 함께 뒤야 자동화가 무너지지 않는다는 점도 함께 배웠습니다.

결국 자동화의 진정한 가치는 데이터를 한곳에 모으는 것이 아니라, **사람이 도구를 의식하지 않고 본연의 업무에 몰입하게 만드는 것**에서 완성된다는 점을 깨달았습니다.

슬랙 점심 주문 요약 봇.

격주 패치 주간마다 반복되는 작은 마찰을 클릭 한 번으로 — 일상의 불편을 그냥 지나치지 않는 자동화

ROLE

기획 · 개발 · 배포

PERIOD

2026.05

TYPE

사내 슬랙 봇 1인 개발 (마이크로 프로젝트)

01 · PROBLEM

점심 주문 하나에 시트와 스크립트가 동원되는 풍경

격주로 돌아오는 패치 주간이면 슬랙 점심 주문 채널에 메뉴 메시지가 올라오고, 팀원들은 숫자 이모지 reaction으로 메뉴를 고릅니다. 문제는 그 다음이었습니다.

시트로 나갔다가 슬랙으로 돌아오는 우회로 — 기존에는 Google Apps Script가 reaction을 읽어 스프레드시트에 집계했는데, 누군가 시트를 열어 스크립트를 수동 실행하고, 그 결과를 다시 슬랙에 손으로 옮겨 적어야 했습니다. '자동화'라고 부르기엔 사람 손이 세 번 들어가는 구조였습니다.

코드에 노출된 사용자 토큰 — 더 큰 문제는 보이지 않는 곳에 있었습니다. 그 Apps Script에는 개인 사용자 토큰이 평문으로 박혀 있었습니다. 점심 주문용 스크립트가 사실은 한 사람의 슬랙 권한 전체를 들고 돌아가는 보안 리스크였던 셈입니다.

02 · APPROACH

슬랙 안에서 시작해서 슬랙 안에서 끝나게

작은 도구일수록 동선이 전부라고 판단했습니다. 시트라는 우회로를 없애고, 주문이 일어나는 바로 그 자리 — 메시지와 스레드 — 에서 모든 것이 끝나는 구조를 목표로 했습니다.

클릭 한 번의 UX — 주문 메시지의 메뉴에서 "주문 요약하기"를 누르면, 몇 초 뒤 같은 메시지 스레드에 메뉴별로 그룹핑된 요약이 댓글로 달립니다. 누가 눌러도 되고, 배울 것도 없습니다.

상태 없는 서버리스 설계 — 외부 DB도, 큐도, 영구 저장소도 두지 않았습니다. Cloudflare Worker 하나가 reaction을 읽어 요약을 만들고, 결과는 슬랙 스레드 자체에 남습니다. 관리할 인프라가 없으니 운영 비용도 0에 수렴합니다.

보안 정상화 — 개인 사용자 토큰 대신 최소 권한 scope 3개만 가진 전용 Bot 토큰을 발급하고, 토큰은 코드가 아닌 Cloudflare Secret으로 관리하도록 바꿨습니다. 기존에 노출돼 있던 사용자 토큰은 폐기를 안내했습니다.

03 · BUILD

작지만 제대로 — 704줄짜리 정석

흐름은 단순합니다: 메시지 단축키 클릭 → Slack이 Worker로 payload 전송 → 서명 검증 후 즉시 ACK → 백그라운드에서 reaction·사용자 정보 조회 → 메뉴별 요약을 스레드에 답글.

작은 프로젝트지만 구조는 큰 시스템과 같은 기준으로 잡았습니다.

- 순수 함수 분리 + TDD — 메뉴 파싱·요약 포맷과 HMAC 서명 검증을 순수 함수로 분리하고 테스트부터 작성했습니다. 텍스트 이모지 코드와 유니코드 혼용, 빈 reaction 같은 엇지 케이스가 모두 테스트로 고정되어 있습니다.
- Slack 보안 요건 준수 — signing secret 기반 HMAC 서명 검증으로 위조 요청을 차단하고, Slack의 3초 응답 제한은 즉시 ACK 후 백그라운드 처리로 해결했습니다.
- 스펙 주도 개발 — 이 규모의 봇도 설계 스펙 → 구현 플랜 → 커밋 단위 구현의 동일한 프로세스로 만들었습니다. 전체 src+test 약 700줄, 커밋 14개.

04 · IMPACT

우회로 하나가 통째로 사라지다

3단계 수작업 → 클릭 1번 — 시트 열기, 스크립트 실행, 결과 옮겨 적기가 메시지 단축키 하나로 대체됐습니다. 주문이 모이는 자리에서 요약까지 끝나니, 결과를 찾으러 갈 곳 자체가 없어졌습니다.

보안 리스크 제거 — 코드에 평문 노출돼 있던 개인 사용자 토큰을 최소 권한 Bot 토큰 + Secret 관리 체계로 교체했습니다.

운영 비용 0의 상시 서비스 — 서버리스 + 무상태 설계로, 배포 이후 관리·비용 부담 없이 패치 주간마다 돌아가는 도구가 됐습니다.

05 · LEARNINGS

자동화 감각은 규모를 가리지 않는다

보이스 워크플로우 시스템이 ‘파이프라인 전체를 다시 설계하는’ 프로젝트였다면, 이 봇은 정반대 끝에 있는 프로젝트입니다 — 모두가 패치 주간마다 겪지만 아무도 문제라고 부르지 않던 마찰을 발견하고, 하루 만에 치우는 일.

작은 도구에서 배운 것이 오히려 선명했습니다. 첫째, **불편은 익숙해지면 보이지 않는다** — 시트를 여는 30초, 결과를 옮겨 적는 1분은 누구도 개선 요청을 올리지 않는 비용이지만, 패치 주간마다 전원이 지불하고 있었습니다. 둘째, **작은 자동화일수록 동선이 전부다** — 기능이 같아도 ‘슬랙 밖으로 나가야 하는 도구’와 ‘쓰레드 안에서 끝나는 도구’는 채택률이 다릅니다. 셋째, **규모가 작다고 기준을 낮출 이유는 없다** — 서명 검증, 최소 권한, 테스트로 고정한 옛지 케이스 같은 기본기는 700줄짜리 봇에서도 그대로 적용했고, 그 덕에 배포 후 손 갈 일이 없는 도구가 됐습니다.

큰 시스템을 설계하는 안목과 일상의 마찰을 줄이는 감각은 결국 같은 근육이라는 것 — 이 작은 프로젝트가 남긴 확신입니다.

라이브 오퍼레이션 및 글로벌 프로젝트 매니징.

AI 솔루션을 현업에 성공적으로 확산시키기 위한 기반, 라이브 게임 생태계에 대한 무결한 이해

ROLE	PERIOD	TYPE
이터널 리턴 개발 PM	2023.04 – Present	라이브 운영 · 글로벌 퍼블리싱 · 외주 매니징

01 · OVERVIEW

역량 정의와 기여 가치

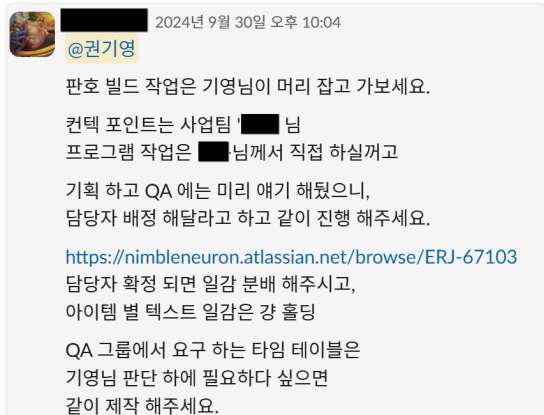
역량 정의 — 글로벌 빌드 총괄, 라이브 크라이시스 통제, 외주 계약 및 행정 오퍼레이션

기여 가치 — 라이브 서비스의 공정과 배포 호흡을 완벽히 파악하여, AI R&D 결과물을 현장에 적용할 때 발생하는 기술적·프로세스적 병목을 최소화함.

02 · TRACK RECORD

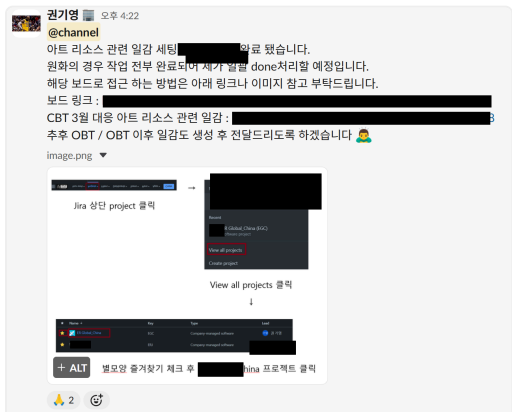
글로벌 빌드와 라이브 운영의 발자취

중국 판호 빌드 메인 개발 PM 총괄 — 전사적 핵심 과제인 중국 서비스 대응 프로젝트의 메인 PM을 맡아, 개발 리드의 전폭적인 신뢰 하에 전체 마일스톤과 배포 파이프라인 관리를 주도했습니다.

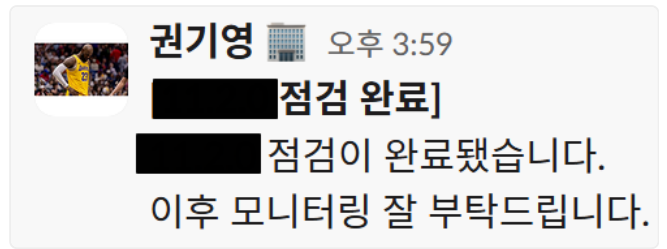
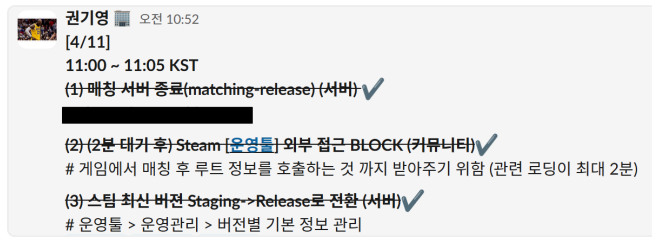


글로벌 퍼블리셔 연계 및 다자간 조율 — 중국 현지 퍼블리셔와 직접 소통하며 요구사항을 정밀하게 조율하는 한편, 사내 사업팀, 프로그램, 기획, QA 등 이종 부서 간의 일감 분배와 병목 제거를 책임졌습니다.

Jira 파이프라인 구조화 — 글로벌 빌드 전용 보드(ER Global_China)를 신설하여 CBT 및 OBT 단계를 위한 대규모 아트 리소스와 텍스트 일감을 체계적으로 구조화하고 관리했습니다.



엔드투엔드 라이브 크라이시스 통제 — 즉각적인 리스크 방어가 요구되는 긴급 핫픽스(Hotfix) 대응부터, 안정적인 콘텐츠 업데이트를 위한 2주 단위의 정규 패치, 그리고 서비스의 판도를 전환하는 3개월 단위의 수많은 시즌 점검 작업을 최전선에서 주도적으로 리딩했습니다.



03 · COLLABORATIONS

대형 글로벌 IP 콜라보레이션 리딩

이터널 리턴의 가장 성공적인 마일스톤 중 하나였던 <길티기어> 및 <페르소나5> 콜라보레이션의 메인 개발 PM을 담당하여 외부 IP 홀더의 피드백 추적 및 내부 제작 일정을 정밀하게 얼라인했습니다. 추가로 이 과정에서 겪은 도구 간 정보 파편화(슬랙, 지라, 드라이브)의 아픔을 뼈저리게 느꼈고, 이는 추후 직접 '콜라보 종합상황판'을 기획 및 구현하여 조직의 협업 인지 비용을 획기적으로 낮출 강력한 도약대가 되었습니다.



04 · OUTSOURCING

철저한 사운드 외주 관리 및 행정 완결

외주 계약 및 정산 전반 총괄 — 개인 크리에이터(사운드 외주자)부터 전문 녹음 스튜디오에 이르기까지, 복잡한 사운드 및 성우 보이스 외주 파이프라인의 계약서 작성 및 세금계산서 발행 등 비용 처리 업무 전반을 리드했습니다.

이해관계자 간 간극 메우기 — 외부 외주 업체의 작업 호응과 사내 개발팀의 리소스 요구사항 사이에서 중간 조율자 역할을 하며, 계약 단계부터 최종 정산까지 잡음이나 법적·회계적 리스크가 발생하지 않도록 철저하게 오퍼레이션을 진행했습니다.

05 · WHY IT MATTERS

아무리 뛰어난 AI 모델이라도 빌드가 배포되고 서버가 돌아가는 라이브 공정의 호흡을 모르면 전사적 확산은 불가능합니다. 글로벌 판호 빌드를 안정적으로 안착시키고 긴박한 라이브 핫픽스 현장을 책임져 온 저의 실전 오퍼레이션 역량은, 엔씨소프트의 AX가 데모와 파일럿에 머물지 않고 게임 개발·라이브 현장의 실제 업무 흐름 속에 정착하게 만드는 가장 확실한 무기입니다.

팀 관리 및 정무적 리더십.

다양한 도메인의 장벽을 허물고, 개성 강한 조직원을 하나의 목표로 정렬시키는 피플 매니지먼트

ROLE	PERIOD	TYPE
People Management · Stakeholder Coordination	2019.01 – Present	피플 매니지먼트 · 부서간 조율 · PD 보좌

01 · OVERVIEW

역량 정의와 기여 가치

역량 정의 — 하이브리드형 조직 케어, 부서 간 갈등 조율, 커뮤니케이션 구조화, 최고 의사결정권자(PD) 보좌

기여 가치 — 조직 내부의 심리적 병목과 갈등을 선제적으로 해소하여, AI 프로젝트 도입 및 프로세스 확산 시 구성원들이 상호 존중과 열린 자세로 협업할 수 있는 토대를 마련함.

02 · APPROACH

국회 정무 감각을 게임 팀 관리에 이식

국회 의원실에서 체득한 소통 노하우 기반의 피플 매니지먼트 — 국회 정책비서 시절 매일 다양한 성향의 민원인 및 정부 기관 관계자들을 대면하며 다져진 깊은 인간 이해도와 소통 노하우를 팀 관리에 이식했습니다. 기획, 밸런스, 캐릭터, 시나리오, 사운드 등 성향과 업무 방식이 완전히 다른 다종다양한 유관 부서 직원들과 밀착 소통하며 팀 내부의 고충을 선제적으로 감지했습니다.

이성적 프로세스와 감성적 케어의 투트랙 대응 — 리소스 배분 및 워크플로우 불일치로 발생한 공정상의 병목은 명확한 '프로세스 수립 및 규칙 제정'을 통해 구조적으로 타파하고, 직무 스트레스로 인한 팀원들의 감정적 지침이 포착될 때는 따뜻한 '인간적인 1:1 심층 상담'을 통해 심리적 마찰을 해소하는 유연한 팀 관리를 전개했습니다.

민심(民心)을 읽던 정무적 감각으로 통제하는 라이브 여론 — 유저 커뮤니티의 동향과 여론의 향방에 따라 프로젝트의 성패가 실시간으로 좌지우지되는 라이브 게임의 특성을 명확히 파악하고 상시 커뮤니티 모니터링을 전개했습니다. 국회에서 실시간으로 민심 동향을 읽어내던 감각을 라이브 오퍼레이션에 그대로 녹여냈습니다.

최고 의사결정권자(PD)를 위한 나침반 역할 — 거센 여론의 파도 속에서 최고 책임자인 PD(프로듀서)가 중심을 잃지 않고 가장 객관적이며 올바른 판단을 내릴 수 있도록, 리스크 요소를 정밀하게 분석하여 데이터 중심의 정무적 보좌역을 전담했습니다. 국회의원의 정책적 의사결정을 가장 가까이서 리스크 없이 보좌했던 경험이 강력한 자산을 세웠습니다.

03 · WHY IT MATTERS

새로운 AI 시스템을 도입할 때 가장 큰 벽은 기술이 아니라 변화를 마주하는 '사람의 거부감'입니다. 성향이 다른 유관 부서원들의 고충을 이성적 프로세스와 인간적 상담으로 해결하고, 거센 여론 속에서도 디렉터의 올바른 결정을 보좌해 온 저의 정무적 리더십은, 엔씨소프트 AX지원실이 전사 AI 전환을 추진하며 마주할 유관 부서 간의 장벽을 허물고 '임직원이 신뢰하고 계속 쓰는 AX 확산'을 이끌어낼 가장 독보적인 자산입니다.