# OVERDARE Senior Software Engineer (AI Agent) 지원 문서

**강태욱**

Email: woogi.dev@gmail.com  Phone: +82 10-9884-3376  Portfolio: https://woogi.is-a.dev
GitHub: https://github.com/woogi-kang

## 자기소개서

OVERDARE의 Senior Software Engineer (AI Agent) 포지션을 보며, 이 역할의 핵심은 단순히 LLM을 제품에 붙이는 것이 아니라 UGC 제작자의 의도를 Studio 안에서 실행 가능한 제작 단위로 바꾸고, 결과물을 검증하며, 다시 개선할 수 있는 AI Agent 경험을 만드는 일이라고 이해했습니다. OVERDARE Studio가 생성형 AI와 Unreal Engine 5를 활용해 모바일 기반 UGC 게임 제작을 쉽게 만드는 플랫폼이라면, AI Agent는 사용자의 자연어 의도와 실제 제작 도구 사이의 가장 중요한 실행 계층이 될 수 있다고 생각합니다.

저는 Flutter 기반 멀티플랫폼 제품 개발에서 출발해 AI Agent Engineering과 AI 업무 자동화까지 확장해 온 Product Engineer입니다. 6년 동안 모바일, TV, 데스크톱, 스마트 글래스, BLE 디바이스, 실시간 음성·자막, 채팅, 현장 운영 자동화처럼 제품 표면이 넓고 예외가 많은 환경에서 문제를 해결해 왔습니다. 최근에는 Claude Code, Codex CLI, Gemini CLI 기반 Agent/Skill/Workflow, MCP/Tool Calling, Structured Output, PromptOps, LLM 평가 하네스, 데이터 수집·검증 파이프라인을 설계하며 AI가 실제 업무와 제품 기능을 실행하는 환경을 만들고 있습니다.

제가 이 역할에 기여할 수 있는 첫 번째 지점은 Agent 실행 환경을 제품 수준으로 다루는 경험입니다. 자연어 요청을 task로 접수하고, 실행 단계를 나누고, 도구 호출과 검증을 거치며, 실패와 산출물을 기록하는 구조를 설계했습니다. 프롬프트를 단순 문자열이 아니라 version, status, metric, rollback이 있는 운영 자산으로 관리했고, A/B evaluation, factual guard, drift guard, human review를 결합해 AI 결과물이 반복적으로 개선될 수 있게 했습니다. 이는 OVERDARE Studio에서 AI Agent가 UGC 제작자의 의도를 해석하고, 제작 액션을 계획하고, 실행 결과를 다시 검증하는 흐름과 직접 연결된다고 봅니다.

두 번째는 제품과 도메인의 복잡한 문제를 끝까지 닫는 실행 경험입니다. 엑스퍼트아이엔씨에서는 AI 스마트 안경 제품군의 Flutter 멀티플랫폼 앱을 리드하며 BLE, GATT, WebSocket, STT, Android Kiosk Mode, OTA, ADB Factory 운영까지 연결했습니다. 프로젝트 기반 컨설턴트로는 OTT, 소셜 주문, 커뮤니티, 숏폼 영상, 채팅, DRM/Video Player, TV 앱, 실시간 동기화 등 콘텐츠와 사용자 상호작용이 많은 제품의 구조 개선과 안정화를 맡았습니다. OVERDARE가 게임 제작, 아바타 커스터마이징, 채팅, 크리에이터 경제를 함께 다루는 플랫폼이라는 점에서, 제품 표면 전체를 보며 기술 문제를 해결해 온 경험을 살릴 수 있습니다.

세 번째는 게임과 UGC 제작 과정에 대한 기반 이해입니다. 저는 게임모바일콘텐츠학과에서 C/C++, WinAPI, OpenGL을 사용한 커스텀 게임 엔진 개발과 Unity 기반 게임 개발을 경험했고, 게임 제작의 기획, 엔진/렌더링, 인터랙션 구현, 테스트, 반복 개선 과정을 전공 단계에서 다뤘습니다. 이 경험은 AI가 생성한 게임 로직, 에셋 배치, 상호작용 규칙을 검증할 때 엔진과 렌더링, 런타임 제약을 함께 보는 기반이 됩니다. 이후 커리어는 앱과 AI Agent Engineering으로 확장되었지만, 게임을 꾸준히 플레이하고 제작 관점에서 바라보며 사용자 조작감, 콘텐츠 루프, 크리에이터 도구의 사용성을 계속 관심 있게 봐왔습니다. AI Agent가 크리에이터를 대신하는 도구가 아니라 크리에이터가 더 빠르게 실험하고 완성도 높은 결과에 도달하도록 돕는 실행 파트너가 되어야 한다는 관점으로 OVERDARE의 문제를 다루고 싶습니다.

OVERDARE에 합류한다면 초기에는 UE5 기반 Studio 도메인을 빠르게 학습하면서, 사용자의 한 문장 요청을 제작 의도, 오브젝트/에셋, 상호작용 규칙, 검증 항목으로 나누는 Agent 실행 계층부터 정리하고 싶습니다. Agent가 어떤 도구를 호출해야 하는지, 실패와 불확실성을 어떻게 사용자에게 드러내야 하는지, 어떤 결과를 회귀 테스트와 human review 대상으로 삼아야 하는지를 함께 설계하겠습니다.

무모하고 대담한 플랫폼을 만든다는 목표는 기술적으로도 제품적으로도 어려운 문제라고 생각합니다. 저는 AI Agent와 제품 엔지니어링을 함께 다룬 경험을 바탕으로, OVERDARE Studio에서 사용자가 더 쉽게 만들고 더 자주 실험하며 더 높은 품질의 UGC를 완성할 수 있는 AI Agent를 만드는 데 기여하고 싶습니다.

## 경력기술서 Executive Summary

저는 제품 엔지니어로서 Flutter 기반 멀티플랫폼 앱 출시와 운영을 6년간 경험했고, 최근에는 Claude/Codex/Gemini CLI 기반 Agent/Skill/Workflow, MCP/Tool Calling, Structured Output, PromptOps, LLM 평가 하네스, 데이터 수집·검증 파이프라인을 설계하며 AI가 실제 업무와 제품 기능을 실행하는 환경을 만들고 있습니다.

OVERDARE의 Senior Software Engineer (AI Agent) 역할과 관련해 가장 강한 경험은 네 가지입니다. 첫째, 자연어 요청을 실행 가능한 task와 tool call로 바꾸고 검증·기록·개선까지 닫는 Agent workflow를 설계한 경험입니다. 둘째, 제품과 조직에 존재하는 다양한 기술 문제를 앱, 백엔드, 데이터, 운영 관점에서 직접 진단하고 해결한 경험입니다. 셋째, 콘텐츠·리포트·데이터 자동화를 PromptOps와 Test & Learn 구조로 운영한 경험입니다. 넷째, C/C++, WinAPI, OpenGL 기반 커스텀 게임 엔진과 Unity 게임 개발 경험을 바탕으로 모바일, TV, 스마트 글래스, 실시간 통신, BLE, 영상·채팅·커뮤니티 제품까지 다루며 UGC/크리에이터 플랫폼에 필요한 제품 감각과 운영 감각을 쌓은 경험입니다.

## Case 1. AI Agent Execution and Evaluation Workflow

AI Agent가 제품이나 조직 업무에서 실제로 쓰이려면 자연어 답변만 잘해서는 부족합니다. 사용자의 요청을 실행 가능한 작업 단위로 나누고, 어떤 도구를 어떤 순서로 호출할지 계획하며, 결과를 검증하고, 실패와 변경 이력을 남겨야 합니다. 특히 UGC 제작 도구의 AI Agent라면 "무엇을 만들고 싶은가"라는 사용자의 의도를 Studio의 구체적인 작업과 제약으로 바꾸는 실행 계층이 중요합니다.

**What I Built**

- Claude Code, Codex CLI, Gemini CLI가 같은 규칙과 skill 자산을 참조할 수 있는 Agent/Skill/Workflow workspace 정리
- 자연어 또는 template 기반 작업 분해 흐름을 만들고 plan/design/develop/QA/deploy/review 단계를 Agent workflow로 분리
- MCP, Tool Calling, Structured Output, Playwright 기반 검증을 조합해 AI 작업 결과를 샘플링 검증, 전수 검증, 자동 승인/보정 흐름으로 연결
- 프롬프트를 version, status, metric, rollback이 있는 운영 자산으로 관리하는 PromptOps 구조 설계

**OVERDARE Application**

- 사용자의 제작 의도를 scene, object, interaction, rule, asset, validation task로 분해
- Agent가 직접 실행할 수 있는 tool contract와 Structured Output schema 정의
- 생성 결과를 품질 기준, 충돌 여부, 누락된 설정, 사용자 의도 부합성으로 검증
- 프롬프트/도구/모델 변경 시 이전 제작 결과와 비교하는 eval set 운영

## Case 2. Product Engineering Across Complex Surfaces

OVERDARE는 UGC 게임 제작, 아바타 커스터마이징, 채팅, 소셜 활동, 크리에이터 경제가 결합된 플랫폼입니다. 이런 제품은 한 가지 기술만 잘해서는 안정적으로 만들기 어렵고, 프론트엔드, 앱, 실시간 통신, 미디어, 운영 도구, 데이터 흐름을 함께 봐야 합니다.

**What I Built**

- AI 스마트 안경 제품군 OWL, C-Biz, C-Sound의 Flutter Mobile/Desktop/Smart Glasses 앱 구조 설계 및 개발 리드
- 전공 과정에서 C/C++, WinAPI, OpenGL 기반 커스텀 게임 엔진과 Unity 게임 개발을 경험하며 게임 제작의 기획, 구현, 테스트, 반복 개선 흐름 이해
- BLE Central/Peripheral, GATT, PIN 인증, MTU 협상, 청크 메시징, 배터리/디스플레이 동기화, BLE 오디오 스트리밍 구현
- 실시간 음성·자막 전달, WebSocket, STT, Opus 기반 기능 구현
- Android Device Owner 기반 Kiosk Mode, OTA update, ADB Factory 웹앱으로 현장 설치·업데이트 자동화
- OTT, TV 앱, 소셜 주문, 커뮤니티, 숏폼 영상, 채팅, DRM/Video Player, 실시간 동기화 프로젝트의 구조 개선과 안정화

**OVERDARE Application**

- Studio와 Agent 사이의 상태 동기화, 실행 결과 표시, 오류 복구 UX 설계
- 사용자 제작 흐름에서 실시간 피드백, 협업, 채팅, 콘텐츠 미리보기 등 상호작용 표면 고려
- 앱/클라이언트/백엔드/운영 도구가 함께 바뀌는 기능을 끝까지 출시·운영

## Case 3. Data Collection, Validation, and Tool-Oriented Automation

Agent는 사용 가능한 도구와 신뢰할 수 있는 데이터가 있을 때 더 강해집니다. 저는 외부 데이터 수집, 검증, 보정, 병합, export까지 닫는 파이프라인을 여러 차례 구축했습니다. 단발성 크롤링이 아니라 중단 후 재개, 실패 재시도, 검증 상태, 산출물 저장까지 포함한 운영 구조를 중요하게 보았습니다.

**What I Built**

- 네이버 플레이스 피부과 4,255건을 대상으로 HTTP-only crawler와 `__APOLLO_STATE__` SSR JSON 파싱 기반 수집 파이프라인 설계
- 서울 1,723건을 6개 구역 split DB로 나누고 SQLite checkpoint, WAL mode, per-record commit, merge, CSV/JSON export 구조 구현
- 동명 병원 문제 해결을 위해 이름 검색, 이름+지역 검색, 후보별 좌표 비교, 도로명 검색 폴백으로 이어지는 matching 구현
- 개인 사이드 프로젝트 CheckYourHospital(병원 홈페이지의 AI 검색/SEO 준비도와 이벤트·프로모션 데이터를 진단·리포트화하는 서비스)에서 locate -> collect -> crosscheck -> R2 upload -> DB insert 흐름의 event pipeline 설계

**OVERDARE Application**

- Studio 내부 도구 호출 결과를 run 단위로 저장하고 재현 가능한 execution trace 구성
- 생성/수정/검증 단계별 checkpoint와 retry 설계
- Agent가 만든 결과물을 rule, asset, dependency, runtime error 관점에서 자동 검증
- 사용자 피드백과 실패 케이스를 eval data로 축적해 Agent 품질 개선에 반영

## Case 4. PromptOps, Test & Learn, and AI Product Quality

LLM 기반 기능은 처음 한두 번 잘 동작하는 것보다 운영 중 품질을 측정하고 개선하는 루프가 중요합니다. 모델, 프롬프트, 도구, 데이터, threshold 중 하나만 바뀌어도 결과 품질이 달라질 수 있기 때문에, 변경과 평가가 함께 움직이는 구조가 필요합니다.

**What I Built**

- prompt versioning, prompt selector, active/testing/archive 상태, A/B evaluator, quality scorer, drift monitor, lesson store 구조 설계
- 콘텐츠 품질과 브랜드 리스크를 줄이기 위해 factual guard, persona guard, human-in-the-loop review, rate limit, emergency halt 적용
- CheckYourHospital에서 AI SEO/AEO 진단, 다국어/의료법/AI 추천 시뮬레이션, PDF 리포트 생성 pipeline 구축

- Memoriz(Flutter/FastAPI 기반 커플 기록 앱)에서 AI album/search 구현 변경 시 eval report script, eval set template, sample report, targeted test 변경 여부를 검사하는 guard script 적용

**OVERDARE Application**

- 사용자 의도 반영도, 실행 가능성, 생성 결과 완성도, 실패 복구율을 eval metric으로 관리
- prompt/tool/model 변경 시 regression case를 돌려 제작 결과가 악화되지 않았는지 확인
- Agent가 모르는 것을 추정하지 않고 사용자에게 확인해야 하는 지점을 UX와 policy로 정의
- 반복적으로 좋은 결과를 만든 pattern을 prompt/skill/tooling에 반영하는 Test & Learn 루프 운영

## Closing

OVERDARE에서 제가 기여할 수 있는 부분은 AI Agent를 "답변하는 기능"이 아니라 "제작을 실행하는 제품 기능"으로 만드는 일입니다. 사용자의 의도를 구조화하고, Studio 도구를 안정적으로 호출하며, 결과를 검증하고, 실패를 학습 데이터로 되돌리는 시스템을 만들 수 있습니다. 동시에 6년간의 제품 엔지니어링 경험을 바탕으로 앱, 백엔드, 데이터, 운영 문제를 함께 보며 OVERDARE Studio의 AI Agent 고도화에 기여하겠습니다.