

# 메타 시니어 엔지니어가 알려주는 하네스 세팅 | 빈 프로젝트에서 하네스 직접 만들기

이 영상은 메타 시니어 엔지니어가 Claude Code 를 활용한 AI 개발에서 '하네스 (Harness)' 의 중요성과 구축 방법을 설명합니다. 시청자는 이미 Claude Code 사용 시 하네스의 기본 개념을 접하고 있음을 인지시키며, 빈 프로젝트에서 직접 하네스 프레임워크를 구축하여 개발 프로세스를 자동화하고 결과물의 품질을 향상시키는 방법을 제시합니다. 특히, CLAUDE.md 를 통한 규칙 정의, docs/ 폴더를 활용한 문서화, 자동 검증 훅 (Hooks) 적용, 그리고 /harness 명령어를 통한 AI 의 단계별 실행 및 자동 커밋 기능이 핵심입니다. 이를 통해 기획부터 실행, 검증까지의 전 과정을 AI 가 주도하여 17 분 만에 빌드 통과 앱을 만드는 데모를 선보이며, 하네스에 입력하는 내용이 결과물의 품질을 결정한다는 교훈을 강조합니다.

## 상위 1% 커스텀 하네스 전략



CHANNEL

실벨개발자

VIDEO ID

AQOVNx87Urs

# Executive Summary

영상 시청 전 빠른 정보 습득을 위한 요약

## SUMMARY

이 영상은 메타 시니어 엔지니어가 Claude Code 를 활용한 AI 개발에서 '하네스 (Harness)' 의 중요성과 구축 방법을 설명합니다 . 시청자는 이미 Claude Code 사용 시 하네스의 기본 개념을 접하고 있음을 인지시키며 , 빈 프로젝트에서 직접 하네스 프레임워크를 구축하여 개발 프로세스를 자동화하고 결과물의 품질을 향상시키는 방법을 제시합니다 . 특히 , CLAUDE.md 를 통한 규칙 정의 , docs/ 폴더를 활용한 문서화 , 자동 검증 훅 (Hooks) 적용 , 그리고 /harness 명령어를 통한 AI 의 단계별 실행 및 자동 커밋 기능이 핵심입니다 . 이를 통해 기획부터 실행 , 검증까지의 전 과정을 AI 가 주도하여 17 분 만에 빌드 통과 앱을 만드는 데모를 선보이며 , 하네스에 입력하는 내용이 결과물의 품질을 결정한다는 교훈을 강조합니다 .

# Video Structure

영상 구성과 논리 흐름

01

하이라이트 및 인트로 (0:00 - 2:01)

02

Part 1: Claude Code 시스템 프롬프트 및 oh-my-claudecode 을 통해 이미 하네스를 사용하고 있음을 설명 (2:01 - 10:56)

03

Part 2: docs, CLAUDE.md, /harness, Hooks 를 활용한 하네스 프레임워크 구축 방법 소개 (10:56 - 18:56)

04

Part 3: FeedbackPulse 앱을 기획부터 실행, 리뷰까지 만드는 라이브 데모 시연 (18:56 - 39:06)

05

마무리 및 핵심 교훈 전달 (39:06 - 끝)

# Key Ideas

정보게시물로 전환할 핵심 아이디어

01

하네스 엔지니어링 (Harness Engineering): AI 개발 프로세스를 구조화하고 자동화하는 프레임워크.

02

Claude Code: Anthropic 의 AI 코딩 도구로, 내장된 시스템 프롬프트에 하네스 개념이 포함되어 있음.

03

oh-my-claudecode: 19 개 에이전트, 파이프라인, 스킴 시스템을 갖춘 범용 하네스 사례.

04

하네스 프레임워크 4 개 레이어: docs/ (PRD, ARCHITECTURE, ADR), CLAUDE.md, harness + execute.py, Hooks.

05

Phase 기반 자동 실행: AI 가 정의된 단계에 따라 작업을 수행하고 상태를 관리하며 자동 커밋하는 방식 (claude-p 헤드리스 모드).

06

자동 검증 Hook: TDD Guard, Dangerous Command Guard, Circuit Breaker 등을 통해 코드 품질 및 안정성 자동 검증.

# DreamLabs Application

DreamLabs 내부 적용 관점

## 01

AI 기반 개발 프로세스 자동화: DreamLabs 내부 프로젝트에서 AI(예: Claude Code)를 활용하여 기획, 코딩, 테스트, 배포의 초기 단계를 자동화하여 개발 효율성을 극대화할 수 있습니다.

## 02

표준화된 프로젝트 구조 및 문서화: docs/ 폴더를 활용한 PRD, 아키텍처, ADR 문서화 표준을 도입하여 프로젝트의 투명성과 유지보수성을 높일 수 있습니다.

## 03

자동화된 코드 품질 및 보안 검증: TDD Guard, Dangerous Command Guard, Circuit Breaker 와 같은 자동 검증 툴을 DreamLabs 의 CI/CD 파이프라인에 통합하여 개발 초기 단계부터 코드 품질과 보안 취약점을 자동으로 검증할 수 있습니다.

## 04

비개발 직군의 AI 활용 확대: 마크다운 기반의 하네스 프레임워크를 통해 기획자나 디자이너도 AI 를 활용한 프로토타이핑이나 간단한 기능 구현에 참여할 수 있는 환경을 조성할 수 있습니다.

## 05

신속한 프로토타이핑 및 아이디어 검증: AI 가 주도하는 17 분 빌드 통과 앱 데모처럼, 새로운 아이디어나 가설을 빠르게 프로토타이핑하고 검증하는 데 이 프레임워크를 활용할 수 있습니다.

# Verification Required

모델 추론 /metadata 한계 / 원본 확인 필요

01

영상에서 제시된 하네스 프레임워크의 실제 코드 구현 방식 및 세부 로직 (GitHub 링크 참조 필요).

02

FeedbackPulse 데모에서 AI가 기획부터 완성까지 수행하는 구체적인 과정과 각 단계별 AI의 역할 및 개입 수준.

03

TDD Guard, Dangerous Command Guard, Circuit Breaker 등 자동 검증 속의 작동 방식 및 실제 적용 효과.

04

claude-p 헤더리스 모드의 실제 동작 방식 및 상태 관리, 자동 커밋 메커니즘.

05

비개발자가 마크다운만으로 하네스를 활용하는 난이도 및 실제 사용 사례.

06

참고: 본 정보는 제공된 YouTube 영상 메타데이터 (제목, 설명, 타임스탬프 등)만을 기반으로 추론되었으며, 실제 영상 내용 및 트랜스크립트를 통한 상세 확인이 필요합니다.

# Source & Download Metadata

게시물과 문서 산출물 추적 정보

## METADATA

Title: 메타 시니어 엔지니어가 알려주는 하네스 세팅 | 빈 프로젝트에서 하네스 직접 만들기  
Channel: 실벨개발자  
Video ID: AQOvNx87Urs  
Source URL: <https://www.youtube.com/watch?v=AQOvNx87Urs>  
Playlist ID: PLHwM6idVO2zyqi2IZeDAiP5QBqRXd2Zyh  
Generated at: 2026-06-13T15:41:46Z  
Source basis: metadata\_and\_model\_inference