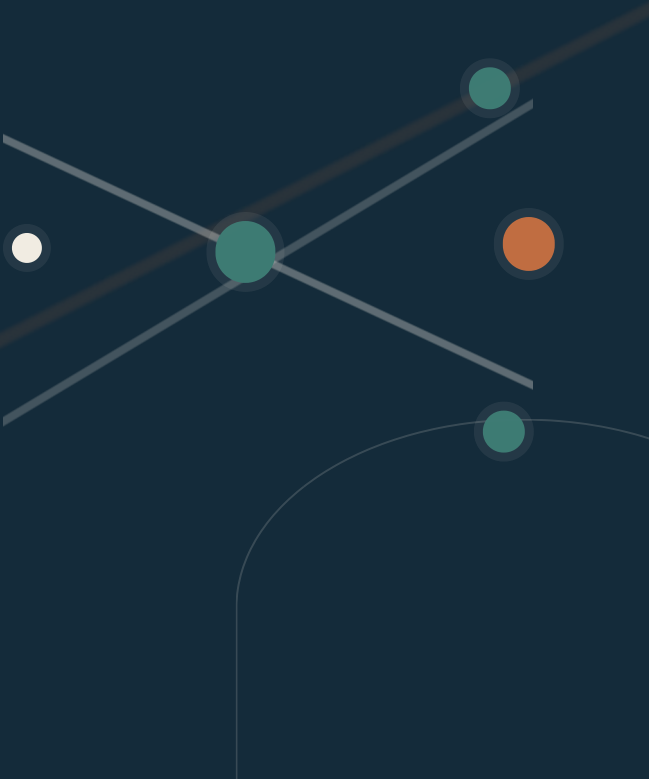




내 일은 AI가 하고, 나는 확인만 한다

1인 기업의
AI 비서 시스템 구축기

지브의 작업실



목차

프로로그 AI에게 일을 맡겼는데, 더 바빠진 이유

1부 AI를 더 썼는데 왜 내가 더 바빠졌을까

1장 일 시킨 사람이 프로젝트 사이를 뛰어다녔다

2장 AI는 도는데 일은 되지 않았다

3장 내가 원한 것은 이런 비서였다

2부 만들었고, 망했고, 뒤집었다

4장 전달만 하는 비서는 필요 없었다

5장 실수가 날 때마다 장치를 붙였다

6장 V2 — 업무판이 보조가 되었다

7장 뒤집기 — 연결을 프로그램에 맡겼다

3부 경계의 설계

8장 지브라는 이름을 붙인 이유

9장 비서가 말을 일과 프로젝트가 말을 일

10장 사람이 완료를 결정해야 하는 이유

4부 두 코딩 에이전트는 무엇이 같고 무엇이 다른가

11장 대화창을 넘어 프로젝트 안에서 일한다

12장 한 업무는 하나의 작업에서 끝낸다

13장 지침과 동료와 절차가 맞는 자리

5부 지브의 안쪽을 열어 보다

14장 요청이 실행 의뢰가 되는 과정

15장 다섯 계층과 두 역할 대화

16장 카드 하나에는 두 대화가 있다

17장 결과는 같은 카드로 돌아오고, 비서는 세 갈래만 고른다

18장 재시작해도 추측해 다시 실행하지 않는다

19장 지침과 기억과 스킬을 어디에 둘까

20장 실제 화면으로 보는 업무판

6부 내 방식으로 다시 설계하기

21장 최소한의 AI 비서실을 종이에 그린다

22장 여섯 단계로 첫 버전을 만든다

23장 정상 흐름과 실패 흐름을 함께 시험한다

24장 어디까지 직접 만들고 어디서 코드를 쓸까

7부 기록이 격차가 된다

25장 일이 끝나도 기록은 남는다

26장 두 번째 엔진이 들어온 날

에필로그 혼자 일하지만 혼자 처리하지 않는다

부록

부록 A 처음 만나는 코딩 에이전트 용어

부록 B 공식 기능과 저자의 설계 경계

부록 C 최소 SQLite 모델과 두 thread binding

부록 D 다섯 종류의 지침과 프롬프트

부록 E 정상·실패 흐름 의사코드

부록 F 테스트 매트릭스

참고 자료

코덱스 공식 자료

클로드 코드 공식 자료

제작과 접근성 자료

내 일은 AI가 하고, 나는 확인만 한다 — 1인 기업의 AI 비서 시스템 구축기

© 2026 지브의 작업실. All rights reserved.

이 전자책은 저작권법의 보호를 받는 저작물입니다. 이 파일은 구매하신 분 본인의 개인적 이용을 위해서만 제공됩니다.

저작권자의 사전 서면 동의 없이 이 책의 전부 또는 일부를 복제·전송·게시·배포·판매·대여하거나 2차적 저작물 작성에 이용하는 것을 금지합니다. 무단 복제·배포·전송 등 저작권 침해 행위는 저작권법 제136조에 따라 5년 이하의 징역 또는 5천만 원 이하의 벌금에 처해지거나 이를 병과할 수 있으며, 별도로 민사상 손해배상 책임이 발생할 수 있습니다.

이 전자책과 소스 코드는 디지털콘텐츠 특성상 제공 개시(다운로드·열람) 후에는 청약철회가 제한됩니다. 구매 전 무료 미리보기로 내용을 확인해 주세요.

이번 개정의 업무판 확인일은 2026-09-07입니다. 이전 운영 사례는 본문의 당시 기록을 따릅니다. 내용 문의와 오류 제보는 지브의 작업실(advercoder.com)로 보내 주세요.

프롤로그

AI에게 일을 맡겼는데, 더 바빠진 이유

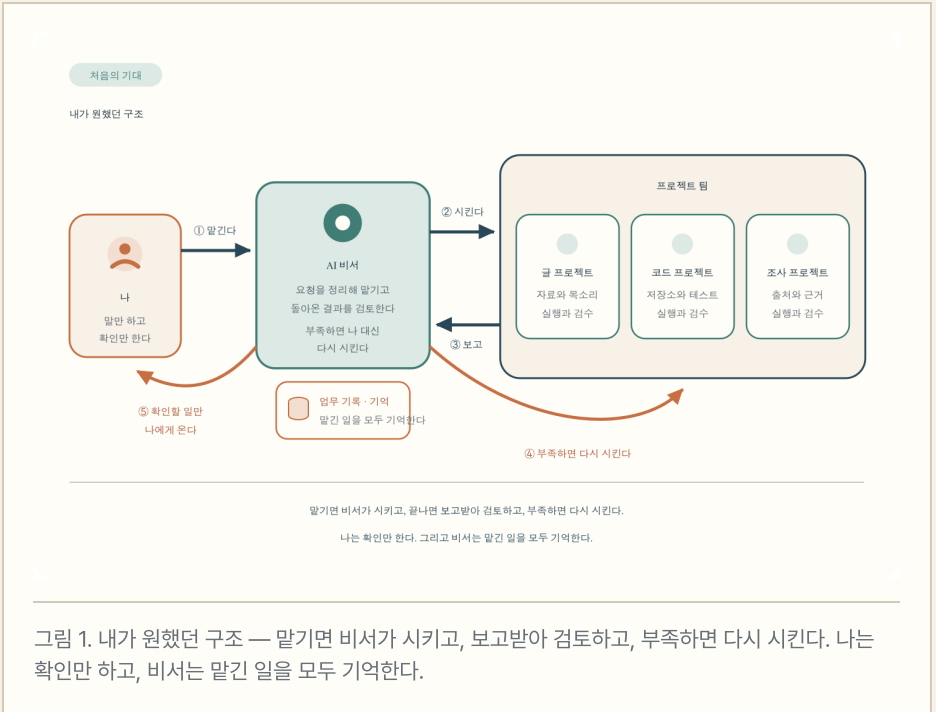
처음에는 AI 프로젝트가 많아지면 내가 할 일이 줄어들 줄 알았다.

한 프로젝트에서는 글을 만들고, 다른 프로젝트에서는 코드를 고치고, 또 다른 곳에서는 자료를 조사했다. 각각의 AI는 자기 자리에서 꽤 많은 일을 해냈다. 문제는 그 사이였다. 어떤 결과가 나왔는지 확인하고, 그 결과를 다음 프로젝트에 설명하고, 빠진 부분을 찾아 다시 지시하는 사람은 나였다. 일을 시킨 사람이 프로젝트 사이를 뛰어다니는 배달원이 된 셈이다.

나는 혼자 일한다. 혼자 일하는 사람이 쓸 수 있는 시간과 집중력에는 분명한 한계가 있다. 그때 이런 생각을 했다.

개인 AI 팀을 만들면 내가 쓸 수 있는 능력의 범위를 넓힐 수 있을까.

내가 처음 머릿속에 그린 모습은 단순했다. 나는 비서 한 명에게만 일을 말한다. 비서는 요청을 정리해 알맞은 프로젝트에 맡긴다. 일이 끝나면 결과와 질문이 다시 비서에게 모이고, 나는 확인하고 결정할 것만 본다.



그때는 이 한 번의 왕복만 만들면 될 줄 알았다.

이 기대가 얼마나 순진했는지, 몇 번을 무너지고 갈아엎은 끝에 어디에 도착했는지가 이 책의 내용이다. 결말을 여기서 다 말하지는 않겠다. 대신 도착한 곳의 풍경만 먼저 보여 주고 싶다.

지금 나는 비서에게 일을 말한다. 그러면 그 일을 아는 프로젝트가 실행한다. 돌아온 결과는 비서가 나 대신 검토하고, 부족하면 나 대신 다시 시킨다. 나는 판단이 필요한 일이 왔을 때만 움직인다. 하루 종일 대화창을 순회하던 사람이, 확인할 일만 확인하는 사람이 됐다.

말로 하면 이렇고, 화면으로 보면 이렇다.



그림 2. 도착한 곳의 풍경 — 진행할 일 13건이 돌아가는 동안, 내가 볼 것은 확인할 일 11건뿐이다. 핸드폰에서 연 내 업무판이다.

그리고 하나 더. 그렇게 지나간 모든 일은 사라지지 않고 업무 카드로 남는다. 무엇을 시켰고, 어떻게 진행됐고, 무엇으로 끝났는지가 쌓인다. 나는 이 기록이 앞으로 같은 AI를 쓰는 사람들 사이의 격차가 될 것이라고 생각한다.

이것은 남이 만들어 준 완성품이 아니다. 내 업무에 맞게 내가 직접 만들었고, 이 책을 따라가면 독자도 자기 업무에 맞게 직접 만들게 된다.

어떻게? 그 답이 이 책의 나머지 전부다.

나는 이 개인 비서에게 ZHIV, 지브라는 이름을 붙였다. 지브는 어떤 제품의 기능 이름이 아니라 내가 설계한 비서 역할에 붙인 이름이다. 책을 덮을 때쯤이면 독자도 자기 비서에게 자기다운 이름을 붙이게 될 것이다.

지금 여러 AI 프로젝트를 돌아다니며 질문에 답하고 결과를 찾고 있다면, 이 책은 그런 사람을 위해 썼다. 어려운 개발 지식은 전제하지 않는다. 낯선 용어는 나올 때마다 풀어서 설명하고, 부록에 처음 만나는 용어 사전도 두었다. 시작은 종이 한 장과 실제 업무 하나면 충분하다.

**AI를 더 고용하기 전에, 먼저 나에게서
무엇을 내려놓고 무엇을 끝까지 붙들지
정해야 한다. 내 AI 비서 시스템도 그
분업표에서 시작했다.**

혹시 이런 이야기가 처음인 독자가 있을지 몰라, 덧붙여 설명하고 싶다. 인공지능을 깊이
쓰다 보면 대부분 이런 상황을 순서대로 만나게 된다.

1 일이 프로젝트 단위가 된다. 프로젝트 안에 코드와 지침과 메모리, 필요한 서브에이전트와 스킬까지 쌓인다. 그 프로젝트 폴더에서 세션을 열어야 어제 하던 일을 이어 갈 수 있다.

2 기다리는 시간이 길어진다. AI가 점점 뛰어나져서 알아서 잘한다. 시켜 놓고 멍하니 있게 된다. 그러다 보면 프로젝트를 하나 더 추가하게 된다. 사이트 하나를 더 만들든, 자동화 하나를 더 붙이든.

3 병목이 나라는 걸 깨닫는다. 어제 시킨 일이 뭐였는지, 애는 어디까지 하고 쉬고 있는 건지, 결과는 왜 이렇게 길어서 읽기 힘든지, 방향이 틀렸는데 대답만 하고 멈춰 있는 건 아닌지 — 전부 내가 쫓아다녀야 보인다.

4 비서가 필요해진다. 나 대신 누가 도와줬으면 한다. 많은 사람이 세션과 1:1로 일한다. 나는 거기서 벗어나고 싶었다.

5 어디서든 접근하고 싶어진다. 굳이 책상 앞에 없어도 되고, 핸드폰에서 음성으로 일을 시키는 쪽이 편하다.

이런 시스템은 존재하지 않았다. 적어도 내가 만들기 전까지, 내가 찾은 범위에는 없었다. 나도 궁금해서 리서치를 했다. 비슷한 시스템이 이미 만들어져 있지 않을까 찾아봤지만 없었다. 에이전트를 대표·팀장 식으로 나눠 구성하는 것들은 많았다. 하지만 프로젝트조차 관리되지 않는 의미 없는 구조가 대부분이었고, 규모가 조금만 커지면 망가질 것이 뻔했다. 프로젝트형으로 효율을 행긴 구조도 있었지만, 실제로는 본인 자료를 연결해 둔 것뿐 별 차이가 없었다. 총괄 비서가 프로젝트들을 관리하는 시스템은 보이지 않았다. 물론 곧 나올 것이다. 아니면 공개되지 않은 곳에서 이미 만들어 쓰고 있는 사람도 있을 것이다.

그런데 코딩 에이전트를 파고들수록, 생각지도 못했던 기능들이 이미 들어 있었다. 나는 이렇게 해석한다. 만드는 쪽도 이런 필요를 미리 내다보고 있었다고. 저장되는 대화와 프로그램이 붙을 수 있는 창구를 이용하면 세션이 세션에게 일을 넘기게 할 수 있고, 서로

작업이 끝났는지도 알 수 있다. 이런 것을 왜 만들어 두었겠는가. 서버에이전트를 단순히 부리는 수준을 넘어, 이제 프로젝트 단위가 서로 소통하면서 일할 수도 있고, 상위 프로젝트를 만들어 서로 연계할 수도 있는 것이다.

시작할 때 내 손에 있던 도구는 코덱스 데스크탑이었다. 지금은 다양한 기능의 터미널과 오픈소스가 쏟아져 나오고 있지만, 나는 아직 전부를 신뢰하지는 않는다. 중간에 멈추는 프로젝트를 많이 봐 왔다. 큰 회사가 자기 이름을 걸고 내놓은 데스크탑과 터미널은 계속 존재할 것이라고 봤다. 지금 내 업무판에는 코덱스와 클로드 코드가 함께 앉아 있고, 카드마다 어느 쪽이 일할지 고른다. 이 책은 둘 중 어느 쪽이 정답이라는 이야기가 아니다. 같은 구조에 무엇을 얹어도 돌아간다는 이야기다.

어쨌든 그래서 나는 이 시스템을 만들기로 했다. 내가 정답은 아니다. 나도 많은 시행착오를 겪었고, 몇 달 뒤에 또 어떻게 바뀔지 모른다. 여러분더러 나를 그대로 따라 하라는 것도 아니다. 나의 시행착오를 딛고, 본인만의 더 좋은 시스템을 만들라는 이야기를 해 주고 싶은 것이다. 나는 이 방향에서 결국 인공지능의 활용이 달라질 것이라 생각한다. 참고로 나는 이 시스템에 별도의 메모리를 구현하지 않았다. 메모리는 각 엔진이 기본으로 제공하는 기능을 그대로 활용하는 중이다. 이 위에 헤르메스처럼 메모리가 뛰어난 모델을 비서로 앉히든, 업무판의 데이터를 LLM 위키로 가공하든, 그것은 여러분의 자유다.

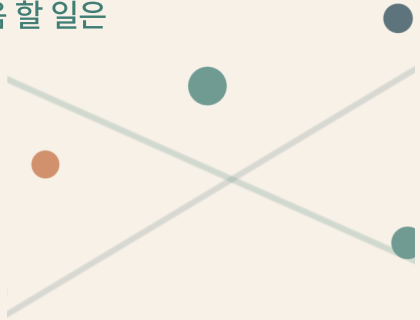
하나만 더. 읽다가 이해가 안 되는 부분이 있으면, 이 책을 AI와 같이 읽기를 권한다. 최대한 쉽게 썼지만, 그래도 익숙하지 않은 부분이 있을 수 있다. 다만 직접 만들어 보다 보면 이해하게 될 것이다.

이 책이 여러분의 AI 인생에 1%의 도움이 된다면 좋겠다.

1부

AI를 더 썼는데 왜 내가 더 바빠졌을까

각 프로젝트는 일했지만, 끝난 결과와 다음 할 일은
내가 기억해야 했다.



1장.

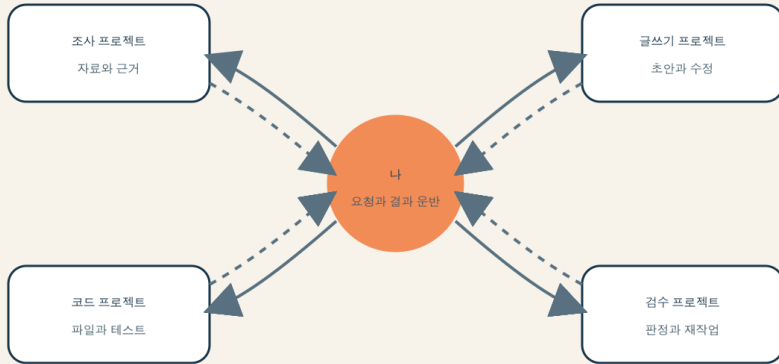
일 시킨 사람이 프로젝트 사이를 뛰어다녔다

AI를 한 곳에서만 쓸 때는 운영 문제가 잘 보이지 않는다. 요청한 자리에서 답을 받고, 마음에 들지 않으면 바로 이어서 수정하면 된다. 하지만 사업에는 서로 다른 종류의 일이 있다. 글은 글의 자료와 목소리를 알아야 하고, 코드는 저장소와 테스트를 알아야 하며, 운영은 실제 계정 상태와 어떤 행동 전에 내 확인이 필요한지 알아야 한다. 일을 제대로 맡기려고 프로젝트를 나누는 순간부터 연결 문제가 생긴다.

내 하루에는 여러 개의 탭이 열려 있었다. 어느 대화에서는 시장 자료를 기다렸고, 다른 대화에서는 전자책 원고를 검토했으며, 또 다른 작업에서는 사이트 수정 결과를 확인했다. AI가 질문을 남기면 내가 찾아가 답했고, 결과가 나오면 다음 프로젝트가 이해할 수 있게 다시 요약했다. 검수에서 문제가 생기면 처음 요청과 결정 내용을 다시 찾아 정리했다.

실행 속도는 빨라졌는데 업무의 전체 속도는 내가 얼마나 잘 기억하느냐에 묶였다. 어떤 일을 먼저 시켰는지, 최신 결과가 어느 것인지, 어떤 수정은 반영됐고 어떤 것은 계획만 세웠는지, 누가 내 답을 기다리는지 모두 내 머릿속에 있었다.

AI가 늘었지만 연결은 사람에게 남았다



각 AI는 일했지만 끝난 결과와 다음 할 일은 내가 기억했다

그림 3. 각 AI는 일했지만 요청·질문·결과와 연결은 사람에게 남아 있었다.

이전에는 이 문제를 AI의 이해력 부족이라고 생각하기 쉬웠다. 더 자세히 지시하거나 더 좋은 모델을 고르면 해결될 것 같았다. 물론 모델의 능력은 중요하다. 하지만 내가 겪은 병목은 다른 곳에도 있었다.

요청이 한 문장의 바람으로 남으면 실행자는 완료를 정의할 수 없다. 필요한 사실이 다른 프로젝트에 있으면 다시 묻게 된다. 결과가 원래 목표와 연결되지 않으면 사람이 어느 결과가 최신인지 찾아야 한다. "완료했습니다"라는 문장이 검증보다 먼저 오면 사용자는 다시 전부 읽어야 한다. 다음 행동을 AI가 마음대로 시작하면 일은 많아지지만 우선순위는 흐려진다.

내게 부족했던 것은 요청을 받는 곳, 각 프로젝트가 책임질 일, 결과를 확인하는 사람, 질문과 결과가 돌아오는 곳을 잇는 구조였다.

모든 일을 한 대화에 몰아넣자 이번에는 오래된 정보가 필요한 정보보다 많아졌다. 코드, 글, 판매, 운영은 각각 자기 파일과 자료 안에서 일해야 했다. 프로젝트를 나누는 것보다 더 중요한 문제는 요청을 받는 곳과 질문·결과가 돌아오는 곳을 정하는 일이었다.

업무를 돌아볼 때 나는 “몇 개의 AI를 썼는가”보다 “몇 번 정보를 다시 옮겼는가”를 보기 시작했다. 조사 결과를 복사했는가. 같은 제약을 다시 설명했는가. 결과가 왔는지 확인하러 다른 대화를 열었는가. 완료 뒤 실제 반영 여부를 다시 물었는가. 이 횟수가 많다면 실행 도구보다 운영 연결을 먼저 손볼 이유가 있다.

한편 프로젝트가 하나이고, 같은 대화에서 파일 하나를 고쳐 바로 검증할 수 있다면 비서실은 필요하지 않을 수 있다. 별도 비서와 업무판을 만드는 비용이 실제 중계 비용보다 크기 때문이다. 구조는 문제의 크기보다 커지면 안 된다.

프로젝트가 이미 여러 갈래이고 그 사이를 직접 오가는 사람이 이 책의 독자다. 그 사람에게 필요한 첫 질문은 “어떤 AI를 더 쓸까”가 아니다.

내 하루의 일 중에서, AI가 해야 할 기억·전달·확인을 지금 내가 대신 해 주고 있는 곳은 어디인가.

결과를 내가 외워 두었다가 다른 프로젝트에 다시 설명하는 일. 어디까지 됐는지 내가 열어 보며 확인하는 일. 이런 자리를 짚을 수 있어야 비서가 앉을 자리가 보인다.

여기까지 읽고 이런 반문이 나올 수 있다.

아니, 그냥 하나씩 시키면 되고, 또 그냥 적당히 판단하라고 시키면 되는 것 아니야? 이렇게 하면 오히려 복잡해지는 것 아니야?

맞다. 하는 일이 단순하고 반복적이고 프로젝트가 늘지 않는다면 굳이 이런 구조까지 만들 필요는 없다.

하지만 AI를 깊이 쓰기 시작하면 프로젝트가 하나둘 늘고, 각 프로젝트에서 새로운 것을 만들고 유지하고 이어 가게 된다. 같은 일을 반복하는 것보다 서로 다른 일을 동시에 진행하는 시간이 많아진다. 그때부터 실행 속도보다 일을 연결하는 방식이 중요해진다.

2장.

AI는 도는데 일은 되지 않았다

일을 시켜 두고 다른 창을 보다가 돌아오면, AI가 대답 한 줄만 남기고 가만히 있는 때가 있다. 다 됐다는 건지, 내 답을 기다리는 건지 화면만 봐서는 알 수 없다. 다시 읽어 보면 AI는 질문을 남겨 두고 멈춰 있었고, 나는 그 질문을 못 본 채 하루를 보낸 뒤였다. 질문 하나에 답을 받지 못해 며칠씩 멈춘 프로젝트를 뒤늦게 발견한 적도 있다.

그렇다면 반대로 생각하게 된다. AI를 쉬지 않고 알아서 돌게 하면 되지 않을까.

해 봤다. 일정한 간격으로 비서를 깨워 상태를 읽게 했다. 실제 변화가 없어도 비서는 카드와 프로젝트를 읽고 “진행 중”이라고 판단했다. 확인하는 AI가 돌고, 그 AI를 감시하는 규칙과 실패 복구까지 붙었다. 시스템은 살아 있는 것처럼 움직였지만 내가 결정할 새로운 정보는 없었다.

방향 없는 가동은 더 나빴다. 밤새 돌려 두고 아침에 보면, AI는 열심히 일했는데 내가 원하는 방향이 아니었다. 멀리 가 있는 결과를 되돌리는 데 다시 시간이 들었다. 토큰은 토큰대로, 시간은 시간대로 사라졌다.

이때 알게 된 사실 하나가 이 책 전체의 출발점이 됐다.

AI를 계속 돌게 하는 것과, 일이 되는 것은 다르다.

빠진 것은 AI의 성능도, 가동 시간도 아니었다. 일이 시작되고, 확인되고, 다음으로 이어지는 길이었다.

3장.

내가 원하는 것은 이런 비서였다

사람 비서를 떠올려 보면 답이 보였다. 사람 비서에게 기대하는 일은 메시지 복사가 아니다. 요청이 모호하면 무엇을 원하는지 정리하고, 담당자가 일을 시작할 만큼 필요한 맥락이 있는지 살피며, 돌아온 결과가 요청과 맞는지 확인한다. 부족하면 다시 보내고, 사람이 결정해야 할 질문만 올린다.

내게 필요한 비서도 같았다.

나 대신 더 정확하게 지시하는 비서. 요청을 받아 목표와 완료 기준으로 바꾸고, 그 일을 아는 곳에 맡기는 비서.

나 대신 검토하는 비서. AI가 결과를 냈을 때 그 결과가 기준에 맞는지 먼저 읽어 보는 비서.

나 대신 다시 시키는 비서. 부족한 부분을 짚어 같은 업무로 재작업을 연결하는 비서. 내가 개입하지 않아도 일이 한 바퀴 더 도는 구조.

그리고 판단할 것만 나에게 가져오는 비서. 내 상황과 업무를 기억하고, 내가 결정해야 할 질문과 확인할 결과만 올리는 비서.

세션은 일이 끝나면 대답하고 멈춘다. 나와 세션 사이에 이런 비서가 있다면, 나는 더 이상 모든 대화창을 순회하지 않아도 될 것 같았다.

그래서 만들어 봤다.

2부

만들었고, 망했고, 뒤집었다

원하는 비서를 그렸으니 만들면 될 줄 알았다.

