

기술

같은 AI를 사용하는데 왜 결과가 이렇게 벌어집니까

같은 도구를 사용합니다. 같은 모델이고 요금제도 같습니다. 하루에 들이는 시간도 비슷합니다. 그런데 몇 달 지나면 나오는 양이 몇 배 차이 납니다. 프롬프트를 잘 쓰는 요령의 문제가 아닙니다. 도구에 무엇을 시키고 있는지가 다릅니다.

방동걸 (주)웨이스 대표이사

2026년 8월 31일

같은 도구를 사용합니다. 같은 모델이고 요금제도 같습니다. 하루에 들이는 시간도 비슷합니다. 그런데 몇 달 지나면 나오는 양이 몇 배 차이 납니다. 웨이스가 VEXPLOR를 개발하며 AI 도구를 매일 쓰는 팀 안에서 확인한 차이를 정리했습니다.

프롬프트를 잘 쓰는 요령의 문제가 아닙니다. **도구에 무엇을 시키고 있는지가 다릅니다.** 한쪽은 오늘 할 일을 도구에게 물어보고 있고, 다른 쪽은 오늘 한 일이 내일 저절로 되게 만들고 있습니다.

이 차이가 눈에 잘 안 띄는 이유가 있습니다. 하루 단위로 보면 둘 다 그날 할 일을 끝냅니다. 겉으로 보이는 하루는 거의 같습니다. 달라지는 것은 그 하루가 끝난 뒤에 무엇이 남아 있느냐이고, 남은 것은 다음 날부터 조용히 일합니다. 그래서 차이가 드러날 때는 이미 몇 배가 벌어진 뒤입니다.

이 차이가 어디서 생기는지, 그리고 뒤쪽을 실제로 어떻게 짜는지 적습니다.



같은 도구에서 시작하는데 남는 것의 규모가 달라집니다

01

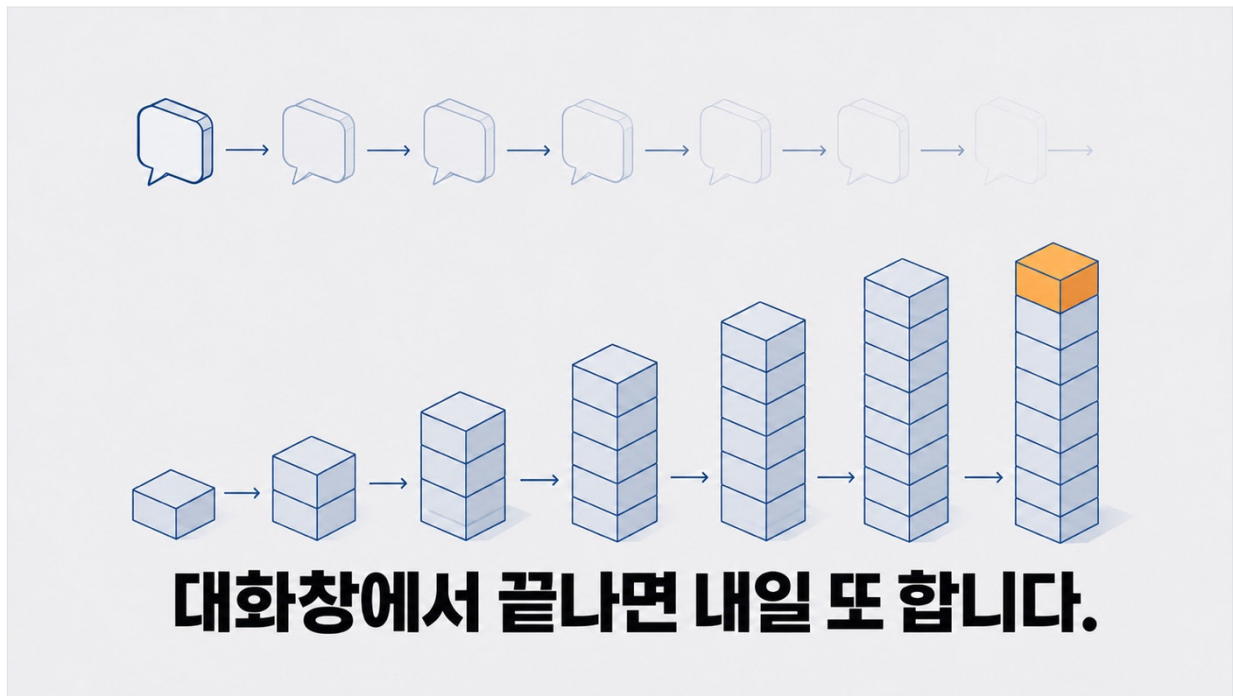
대화창에서 끝나면 내일 같은 일을 또 합니다

첫 번째 방식은 이렇게 생겼습니다. 일이 생기면 창을 열고, 물어보고, 답을 받아 쓰고, 창을 닫습니다. 다음에 비슷한 일이 오면 다시 창을 열고 다시 물어봅니다.

이 방식도 효과가 있습니다. 검색보다 빠르고 초안이 나옵니다. 문제는 **효과가 그날로 끝난다는 것**입니다. 어제 한 일이 오늘을 도와주지 않습니다.

두 번째 방식은 같은 일을 한 번 처리한 뒤 그 처리 과정 자체를 밖에 남깁니다. 다음에 같은 일이 오면 남겨 둔 것이 처리하고, 사람은 결과만 봅니다. 한 번에 아끼는 시간은 얼마 안 됩니다. 그런데 이게 매일 쌓입니다.

방식	오늘 얻는 것	한 달 뒤
물어보고 닫는다	그 일 하나	같은 자리
처리를 남긴다	그 일 하나 + 남은 것	안 해도 되는 일이 늘어남



한쪽은 흔적이 남지 않고 다른 쪽은 매일 쌓입니다

한 달로 보면 차이가 별로 안 나 보입니다. 반년이 지나면 한쪽은 여전히 하루치 일을 하고 다른 쪽은 하루치 일을 하면서 지난 반년 동안 만든 것들이 동시에 돌아옵니다.

02

업무를 자동화한다는 것이 실제로 무엇인가

"자동화"라는 말이 크게 들려서 대단한 것을 만드는 일처럼 느껴지는데, 실제로 하는 일은 네 가지입니다.

첫째, 같은 판단을 반복하는 자리를 찾습니다. 자동화할 수 있는 것은 반복되는 작업이 아니라 반복되는 판단입니다. 매번 손으로 하지만 할 때마다 같은 기준으로 정하고 있는 자리가 있습니다. 그 자리가 후보입니다. 반대로 매번 다르게 정하는 자리는 아직 자동화 대상이 아닙니다.

둘째, 그 판단 기준을 머리 밖으로 꺼냅니다. 여기가 실제 작업의 대부분입니다. "적당히 보고 정한다"를 문장으로 적으면 대개 조건이 서너 개 나오고, 그중 하나는 본인도 몰랐던 조건입니다. 적어 놓고 나면 그제야 시킬 수 있는 형태가 됩니다.



판단 기준을 머리 밖으로 꺼내 두는 것이 실제 작업의 대부분입니다

셋째, 결과가 맞는지 확인할 수단을 따로 만듭니다. 이걸 빼면 자동화가 아니라 자동 오류 생산입니다. 뒤에서 따로 적겠습니다.

넷째, 다른 사람이 쓸 수 있는 자리에 둡니다. 내 컴퓨터에서만 돌아가면 그건 내 개인 요령이지 회사가 가진 것이 아닙니다. 이 단계까지 가야 옆 사람의 일도 같이 줄어듭니다.

03

대부분 두 번째에서 멈춥니다

네 단계 중 사람들이 실제로 걸리는 자리는 거의 항상 둘째입니다.

이유가 있습니다. 오래 해서 몸에 붙은 판단은 **본인도 언어로 갖고 있지 않습니다**. 숨기는 게 아니라 꺼내 본 적이 없어서 그렇습니다. "그건 상황 봐서 하는데요"라는 답이 나오면 그 사람이 불성실한 게 아니라 그 판단이 아직 말의 형태가 아닌 것입니다.

여기서 막히면 두 방향으로 갑니다. 한쪽은 "이건 자동화가 안 되는 일이다"로 결론 내고 돌아갑니다. 다른 쪽은 자기가 그 판단을 실제로 내린 사례를 열 개쯤 늘어놓고 거기서 공통점을 찾습니다. 원칙을 먼저 적으려 하면 안 나오는데, 사례를 늘어놓으면 나옵니다.

사례에서 기준을 뽑는 것이 어떻게 되는지 하나만 적겠습니다. 들어온 문서를 보고 처리 순서를 정하는 일이 있다고 하면, 원칙을 물었을 때는 "급한 것부터 한다"가 나옵니다. 이 문장으로는 아무것도 시킬 수 없습니다. 그런데 지난주에 실제로 처리한 다섯 건을 늘어놓고 각각 왜 그 순서였는지 적어 보면, 급한 정도만 본 게

아니라는 게 드러납니다. 밖으로 나가는 것인지, 되돌릴 수 있는지, 다른 사람이 기다리고 있는지가 함께 걸려 있습니다. 원칙은 한 줄인데 실제 기준은 네 줄이고, 시킬 수 있는 것은 네 줄 쪽입니다.

이 차이가 앞에서 말한 격차의 실제 내용입니다. 도구를 다루는 실력이 아니라 **자기가 무엇을 하고 있는지를 꺼내 볼 수 있는가**입니다. 그래서 이 능력은 도구가 바뀌어도 남습니다. 모델이 바뀌면 프롬프트 요령은 새로 익혀야 하는데, 자기 판단을 문장으로 꺼내 둔 것은 그대로 다음 도구에 넘어갑니다.

04

확인을 같은 도구에 맡기면 확인이 아닙니다

셋째 단계가 왜 빠지면 안 되는지 적겠습니다.

만든 것이 맞게 돌아가는지 확인해야 하는데, 확인하는 방법을 만든 것과 같은 곳에서 가져오면 확인이 성립하지 않습니다. 답을 준 쪽에 "이거 맞아?"라고 되물어보는 것, 같은 방식으로 한 번 더 돌려 보는 것, 다른 표현으로 다시 시켜 보는 것 — 셋 다 확인한 느낌은 나는데 확인이 아닙니다. **판정 수단이 판정 대상 안에 있습니다.**

이게 실제로 어떻게 터지는지 몇 가지 적습니다. 저희가 밟은 것들입니다.

- **검사는 통과인데 검사가 대상에 닿지 않았습니**다. 검사기가 잘못된 경로를 보고 있었고, 볼 것이 없으니 실패할 것도 없어서 계속 통과가 났습니다.
- **정상 응답을 받았는데 화면에는 아무것도 없었습니다.** 서버는 성공을 반환했고 자동 점검도 전부 통과였는데, 실제 페이지 본문이 통째로 비어 있었습니다. 사람이 열어 보고서야 알았습니다.
- **내 컴퓨터에서만 통과였습니다.** 예전에 만들어져 남아 있던 파일이 검사를 대신 통과시키고 있었고, 그 파일이 없는 곳에서는 처음부터 실패였습니다.

셋의 공통점은 **자동 점검이 전부 초록불이었다**는 것입니다. 그래서 확인 수단은 만든 것과 다른 경로여야 합니다. 직접 계산해 보기, 원본 기록을 열어 보기, 실제 화면을 사람이 보기, 다른 방법으로 같은 값을 한 번 더 만들어 보기. 느리고 재미없는데 이게 없으면 자동화가 늘수록 틀린 것이 빨리 늘어납니다.



확인 수단이 확인 대상 안에 있으면 통과 신호는 아무것도 보증하지 않습니다

05

자동화하면 안 되는 자리도 있습니다

반대쪽도 적어야 균형이 맞습니다. 전부 넘기려는 사람과 하나도 안 넘기는 사람이 똑같이 위험합니다.

넘기지 말아야 할 자리는 세 가지로 갈립니다.

자리	왜 안 되는가	대신 할 것
매번 다르게 정하는 일	기준이 아직 없다	사례를 더 모은다
틀리면 못 되돌리는 일	확인이 늦게 온다	사람이 마지막에 본다
판정 자체가 목적인 일	대상과 수단이 겹친다	다른 경로로 켜다

두 번째가 특히 그렇습니다. 밖으로 나가는 문서, 돈이 움직이는 결정, 사람에 대한 판단은 잘못돼도 나중에 알게 되고 그때는 이미 나가 있습니다. 이런 자리는 앞 단계까지만 넘기고 내보내는 동작은 사람이 합니다. 저희도 문서는 자동으로 만들지만 보내는 버튼은 사람이 누릅니다.

여기서 자주 나오는 오해가 하나 있습니다. 이런 자리를 남겨 두는 것이 자동화를 덜 한 상태라고 생각하는 것인데, 반대입니다. 어디까지 넘길지 정해 둔 상태가 아무 데나 넘기는 상태보다 앞서 있습니다. 선을 못 그은 사람은 사고가 나면 그다음부터 전부 손으로 돌아갑니다.

06

일이 줄어드는 게 아니라 종류가 바뀝니다

자동화를 늘리면 한가해질 것 같은데 그렇게 안 됩니다.

손으로 하던 일이 빠지는 대신 **만든 것들을 돌보는 일**이 생깁니다. 조건이 바뀌면 고쳐야 하고, 어긋나면 왜 어긋났는지 찾아야 하고, 새로 들어온 사람이 쓸 수 있게 설명해야 합니다. 총량은 크게 안 줄어드는데 내용이 달라집니다.

그래서 하루가 이렇게 바뀝니다. 예전에는 그 일을 직접 했고, 지금은 **그 일이 되게 만드는 구조를 짜고 그것이 제대로 돌고 있는지 봅니다**. 만들어 놓은 것이 늘어날수록 뒤쪽 비중이 커집니다.

이게 편한 변화는 아닙니다. 직접 하면 오늘 끝났다는 느낌이 확실한데, 구조를 짜는 일은 오늘 뭘 했는지가 잘 안 보입니다. 대신 몇 달 뒤에 보입니다.

넷째 단계에서 걸리는 것도 여기서 나옵니다. 만든 것을 남이 쓰게 하려면 **그 사람이 알아야 할 것이 그 안에 들어 있어야 합니다**. 무엇을 전제하고 있는지, 어긋나면 어디를 봐야 하는지, 손대면 안 되는 부분이 어디인지. 이걸 안 적어 두면 만든 사람 말고는 아무도 못 고치고, 그 사람이 자리를 비우는 순간 팀이 멈춥니다. 개인 요령을 조직이 가진 것으로 만드는 비용이 여기 붙습니다.

그래서 잘 굴러가는 팀을 보면 만든 개수보다 **남이 이어받은 개수**가 다릅니다. 혼자 열 개 만든 사람과 다섯 개 만들고 그중 넷을 남이 쓰게 만든 사람 중에 뒤쪽이 앞섭니다. 앞쪽은 본인이 병목이 되어 있고 뒤쪽은 아닙니다.

07

그래서 무엇부터 하면 됩니까

한 번에 다 바꾸려고 하면 안 됩니다. 반복되는 판단 하나만 고르는 게 시작입니다.

1. 이번 주에 세 번 넘게 같은 방식으로 정한 일을 하나 고릅니다
2. 그 판단의 실제 사례를 다섯 개 적습니다 — 원칙 말고 그날 있었던 일로
3. 다섯 개에서 공통 조건을 뽑아 문장으로 만듭니다
4. 그 문장대로 시켜 보고, **결과를 다른 방법으로 확인합니다**
5. 되면 옆 사람이 쓸 수 있는 자리에 둡니다

한 개 만드는 데 반나절쯤 걸리고 아끼는 시간은 하루에 몇 분입니다. 하나만 보면 남는 장사가 아닌 것처럼 보입니다. 이걸 계속하는 사람과 안 하는 사람의 차이가 반년 뒤에 몇 배로 나타납니다.

조직 전체가 이렇게 일하기 시작하면 무슨 일이 생기는지는 업무를 전부 AI와 함께 하면 조직에서 무엇이 달라집니까에 따로 적었습니다.

이번 주에 같은 판단을 세 번 넘게 반복하신 자리는 어디였습니까?

WACE(웨이스)는 공장이 스스로 판단하고 멈추지 않게 하는 제조 AI 운영체제 VEXPLOR를 만드는 회사다.



원문 blog.vexplor.com/post/why-the-same-ai-gives-different-results

발행 2026년 8월 31일 · 방동걸 (주)웨이스 대표이사



원문



홈페이지



링크드인

회사 주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 ·
사업자등록번호 439-87-02852

주소 (18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동)
1332·1333·1334·1335호 (영천동, 실리콘밸리동탄)

연락처 전화 070-4730-0629 · contact@wace.me

홈페이지 wace.me · linkedin.com/company/wace-inc