비즈니스

저희가 구축한 시스템에서 수정 요청 응답이 9일에서 18분이 됐습니다

방동걸 (주)웨이스 대표이사 2026년 8월 25일

숫자부터 보여 드리겠습니다. 저희가 고객사에 구축해 드린 시스템의 수정·기능추가 요청을 AI가 직접 개발해 응답하는 체계를 30일 넘게 운영했고, 그 기간 실제 응답 119건을 원장에서 전수 집계했습니다. 실제 응답 119건의 **절반이 18.4분 안에** 첫 응답을 받았습니다. 같은 원장에 쌓여 있는 과거 2년치 사람 처리 247건은 **절반이 9일을 넘겨서야** 답을 받았습니다. VEXPLOR 개발사 웨이스가 고객사 시스템을 직접 운영하며 집계한 숫자입니다.

이 글은 그 숫자가 어떻게 나왔고, 빠르기만 한 것은 아닌지 무엇으로 확인했으며, 고객 입장에서 실제로 무엇이 달라지는지를 정리한 실측 보고입니다.

01

유지보수가 느린 진짜 이유는 실력이 아니라 대기열입니다

구축이 끝난 시스템에도 수정 요청은 계속 생깁니다. 검색이 안 되는 품목, 화면에 하나 더 있었으면 하는 칸, 새 업무가 생겨 필요해진 기능. 하나하나는 몇 시간짜리 일입니다.

그런데 실제 처리 기록을 보면 사람 처리의 평균은 37.7일이었습니다. 절반 기준 9일과 평균 37.7일의 차이가 말해 주는 것이 있습니다. **오래 걸린 건은 어려워서가 아니라, 급하지 않다고 분류되어 대기열 뒤로 밀렸기 때문**입니다. 개발자는 늘 더 급한 일을 먼저 합니다. 그 판단 자체는 합리적인데, 결과적으로 고객의 "사소한" 요청은 몇 주씩 기다리게 됩니다.

평균 37.7일의 실체를 뜯어 보면 더 분명합니다. 처리에 37일이 걸린 것이 아니라, **36일 기다리고 하루 만에 처리된 것**에 가깝습니다. 장애 대응과 신규 구축과 "급하다고 전화 온 건"이 계속 끼어들고, 급하지 않다고 분류된 요청은 새로 들어오는 급한 일에 반복해서 추월당합니다. 대기열의 문제는 게으름이 아니라 우선순위 배분의 필연적 결과라서, 사람을 더 재촉한다고 풀리지 않습니다.

그래서 저희가 바꾼 것은 개발 실력이 아니라 **대기열이 생기는 구조 자체**입니다.

02

요청이 들어오면 AI가 바로 개발을 시작합니다

저희 고객사는 관리 시스템에 수정 요청을 등록합니다. 지난 30일 넘는 기간 동안, 새로 들어온 요청은 사람의 분류나 배정을 기다리지 않고 AI가 바로 받아 개발을 시작했습니다.

흐름은 이렇습니다.

- **접수** — 고객이 요청을 등록하면 AI가 즉시 받아 내용을 파악합니다.
- **개발** — AI가 실제 코드를 수정하고, 개발 서버에 반영합니다.
- **확인 요청** — 고객이 개발 서버에서 결과를 직접 확인할 수 있게 알려 드립니다.
- **운영 반영** — 고객이 승인하면, 담당 개발자가 확인한 뒤 운영 서버에 반영합니다.



고객 요청이 접수부터 개발 서버 반영까지 자동으로 흐르고, 운영 반영 앞에 사람 승인이 있습니다.

마지막 단계를 눈여겨봐 주시기 바랍니다. **운영 서버 반영은 자동이 아닙니다.** 고객이 승인하고, 담당 개발자가 확인한 뒤에 반영됩니다. 공장에서 실제로 돌아가는 시스템을 다루는 일이라, 빠른 것보다 중요한 것이 있다고 판단했습니다. 자동인 구간과 사람이 확인하는 구간의 경계를 명확히 두는 것이 저희가 이 체계를 설계한 방식입니다.

03

18분 안에 오는 응답에는 무엇이 담겨 있을까요

응답 속도 이야기를 하면 "자동 접수 문자 아니냐"는 질문을 받습니다. 아닙니다. 도착하는 것은 처리 결과입니다.

실제 사례를 하나만 일반화해 보여 드리면 이렇습니다. "품목 검색창에 띄어쓰기로 두 단어를 넣으면 검색이 안 된다"는 요청이 들어왔습니다. AI는 검색 로직을 고치고 개발 서버에 반영한 뒤 이렇게 답했습니다 — 어떤 화면의 어떤 동작을 고쳤는지, 이제 어떻게 동작하는지, 어디에서 확인하면 되는지. 고객은 개발 서버에서 바로 눌러 보고 승인 여부를 정합니다.

- **수정 내용 설명** — 무엇을 어떻게 바꿨는지를 요청자의 말로 다시 설명합니다. 코드 이야기가 아니라 화면과 동작 이야기입니다.
- **확인 경로** — 고칠 결과가 반영된 개발 서버 주소와 확인 방법이 함께 옵니다. "믿어 달라"가 아니라 "직접 확인해 달라"는 구조입니다.
- **승인 단추** — 확인 후 승인하면 담당 개발자가 검토를 거쳐 운영 서버에 반영합니다. 반려하면 그 사유가 다음 수정의 입력이 됩니다.

04

30일 실측, 세 가지 숫자로 요약됩니다

지표	AI 자동 개발	사람 처리
절반이 답 받기까지	18.4분	9일
24시간 내 응답	69%	19%
14일 내 재접수율	1.7%	2.4%

- **절반이 18.4분 안에** — 실제 응답 119건 기준입니다. 평균은 39.1시간으로 이보다 훨씬 큰데, 소수의 복잡한 건이 평균을 끌어올리기 때문입니다. 그래서 저희는 평균만 따로 인용하지 않고 항상 절반 기준과 함께 봅니다.
- **24시간 내 응답 69%** — 사람 처리는 같은 기준으로 19%였습니다. 하루 안에 답을 받는 경험과 다음 주에 받는 경험의 차이입니다.
- **재접수율 1.7%** — 빠른 응답이 부실한 응답 아니냐는 질문에 대한 저희의 답입니다. 응답 후 14일 안에 같은 고객사가 비슷한 사안으로 다시 요청을 올린 비율이 AI 처리에서 1.7%, 사람 처리에서

2.4%였습니다. **속도를 얻는 대신 품질을 잃은 것은 아닌지를, 같은 원장의 같은 기준으로 재서 확인했습니다.**



대기열에 쌓이던 요청과 몇 분 만에 돌아오는 응답의 대비입니다.

05

빠른 응답이 만드는 진짜 변화는 요청의 성격이 바뀌는 것입니다

몇 주 걸리던 것이 몇 분이 되면, 처음에는 속도에 놀라지만 조금 지나면 다른 변화가 생깁니다. **고객이 요청을 참지 않게 됩니다.**

응답이 몇 주 걸리는 환경에서 고객은 요청을 고릅니다. "이건 불편하지만 요청할 정도는 아니야"라며 넘어가는 것들이 쌓입니다. 시스템은 멀쩡히 돌아가는데 현장의 작은 불편은 그대로 남는 상태가 됩니다.

응답이 몇 분 단위가 되면 그 계산이 바뀝니다. 검색창이 띄어쓰기를 못 알아듣는 것 같은, 예전이라면 말하지 않고 넘어갔을 요청이 들어옵니다. 그리고 그날 해결됩니다. 시스템이 실제 사용하는 사람들의 손에 점점 맞춰지는 속도가 달라집니다.

구축 업체 입장에서도 달라지는 것이 있습니다. 유지보수 인력이 단순 수정 대기열을 비우는 데 쓰이지 않으니, 사람은 판단이 필요한 일에 집중합니다. 위 흐름에서 사람이 맡는 자리가 "모든 요청의 처리"에서 "운영 반영 전 확인"으로 옮겨 간 것이 그 결과입니다.

06

아직 다 이룬 것은 아닙니다

있는 그대로 적겠습니다.

- **모든 요청이 몇 분 만에 끝나는 것은 아닙니다.** 복잡한 건은 하루 이상 걸리고, 사람 팀으로 넘겨 처리한 사례도 있습니다. 범위 밖의 요청을 억지로 자동 처리하지 않는 것도 설계의 일부입니다.
- **측정을 계속 정교하게 만들고 있습니다.** 응답 속도와 재접수율 외에, 해결 확정 기준과 단계별 처리 시간을 더 촘촘히 재는 작업을 진행 중입니다. 90일 시점에 다시 집계해 공유하겠습니다.
- **이 수치는 저희 고객 환경의 실측입니다.** 모든 시스템, 모든 요청 유형에 같은 숫자가 나온다고 일반화할 수는 없습니다.

그래도 한 가지는 분명해졌습니다. 유지보수 응답이 며칠씩 걸리는 것은 자연법칙이 아니라 대기열 구조의 결과이고, 그 구조는 바꿀 수 있습니다.

07

같은 것을 여러분의 기록으로도 재 보실 수 있습니다

저희가 쓴 방법은 특별한 도구가 필요 없습니다. 최근 수정 요청 10건을 꺼내서 두 시각만 적으면 됩니다 — 요청을 보낸 시각과, 접수 안내가 아닌 **실제 처리 결과**가 온 시각.

- 10건을 빠른 순서로 줄 세웠을 때 **다섯 번째 건이 걸린 시간**이 그 조직의 체감 응답입니다. 평균은 극단값 한두 건에 끌려다니므로 이 값이 더 정확합니다.
- **한 달을 넘긴 건이 몇 건인지** 세어 보시면 대기열이 얼마나 쌓여 있는지 보입니다.
- 유지보수 업체와 계약을 검토하신다면 응답 속도 말고 **재접수율**(처리 후 같은 문제로 다시 접수되는 비율)과 **대기 구간**(요청이 어디서 며칠 머무는지)을 물어보시기 바랍니다. 속도 숫자는 만들기 쉽지만, 이 둘은 기록이 없으면 답하지 못합니다.



요청 등록 시각과 실제 처리 결과가 온 시각, 두 개만 있으면 같은 측정을 할 수 있습니다.

웨이스가 하는 일이 궁금하시면: wace.me

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

원문 blog.vexplor.com/post/maintenance-response-18-minutes
발행 2026년 8월 25일 · 방동걸 (주)웨이스 대표이사

회사 주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 · 사업자등록번호 439-87-02852
주소 (18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동) 1702-1708호 (영천동, 실리콘앨리동탄)
연락처 전화 070-4730-0629 · contact@wace.me
홈페이지 wace.me · linkedin.com/company/wace-inc

원문 홈페이지 링크드인