인사이트

어긋난 재고 숫자를 고치는 대신, 어긋나지 않는 규칙을 넣었습니다

방동걸 (주)웨이스 대표이사 2026년 8월 27일

앞선 글에서 재고 숫자가 어긋나는 자리 네 곳을 적었습니다. 소요량 계산이 원본 표를 잘못 고른 자리, 같은 출하가 화면에 따라 다르게 기록된 자리, 문서번호 하나가 두 가지를 뜻하던 자리, 장부끼리 대사가 안 되던 자리입니다. VEXPLOR로 공장 시스템을 구축하는 웨이스가 그 네 곳을 실제로 고치며 정리한 후속 글입니다.

전부 저희가 만들고 있는 생산관리 소프트웨어의 시험 데이터에서 나왔습니다. 넷 중 둘은 고쳤고 둘은 아직 열려 있는데, 고친 이야기만 적으면 이 글은 쓸모가 없습니다. **다음에 같은 모양이 다시 생기지 않게 무엇을 규칙으로 만들었는지**가 남는 이야기라서 그쪽을 적습니다.

01

고치는 것과 다시 안 생기게 하는 것은 다른 일입니다

앞의 둘은 고치는 데 하루가 걸리지 않았습니다. 읽는 표를 바꾸고, 빠진 함수 호출을 넣으면 끝입니다.

뒤의 둘은 아직 열려 있습니다. 문서번호는 접두어를 바꾸지 않았습니다. 정의가 선언해 둔 체계라 저희 판단으로 손대면 체계가 갈라지기 때문이고, 대신 어긋남을 잡아내는 판정을 하나 새로 넣었습니다. 장부 대사는 차이를 찾기만 했고, 매달 자동으로 도는 자리는 아직 없습니다.

문제는 이 넷이 **서로 다른 사람이 서로 다른 날에 만든 결함**이라는 점입니다. 각자 그 순간에는 그럴 만한 선택이었습니다. 소요량 계산을 쓴 사람은 재고가 있는 표를 읽었을 뿐이고, 출하 화면을 만든 사람은 저장이 되는지 확인했을 뿐입니다. 아무도 실수하지 않았는데 결과가 어긋났다면, 고쳐야 하는 것은 사람이 아니라 **그 선택을 혼자 하게 내버려 둔 자리**입니다.

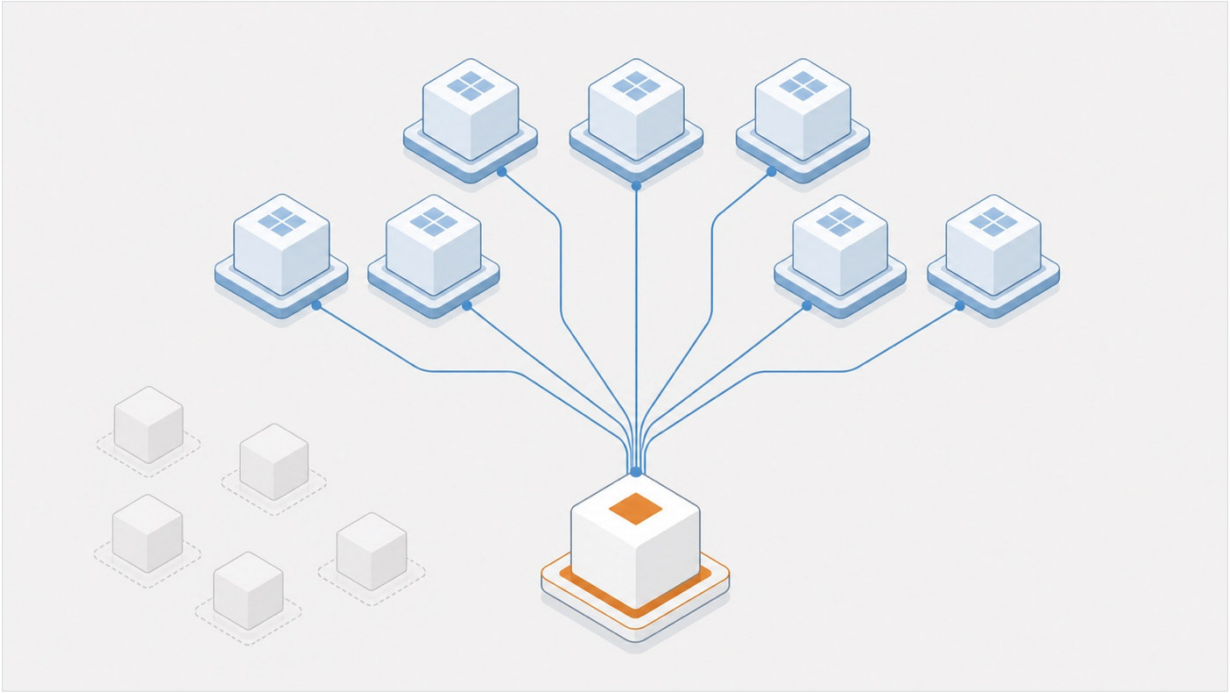
그래서 이 넷을 다루면서 규칙을 여섯 개 넣었습니다. 그중 다섯은 자동 판정 목록에 새 항목으로 들어가 있어 어기면 배포 전에 걸리고, 나머지 하나는 표기 규칙입니다.

02

첫째, 원본이 어느 표인지 선언에 적습니다

소요량 계산이 창고 원장을 읽고 있던 것이 왜 아무에게도 안 걸렸는지를 되짚어 보니, 답이 허탈했습니다. **어느 표가 원본인지가 어디에도 적혀 있지 않았습니다.** 코드가 하나를 골랐고, 그 선택이 문서로 올라온 적이 없습니다.

그래서 값을 읽는 쪽이 아니라 **정의하는 쪽**에 원본을 적게 했습니다. "가용 재고의 원본은 회계 장부이고, 창고 원장은 실물 위치를 담당한다"가 선언에 문장으로 들어가 있어야 합니다. 선언에 없으면 계산이 그 값을 읽지 못합니다.



원본을 한 곳에 선언해 두면 모든 계산이 그 하나를 지납니다. 흩어져 있던 원본 후보는 연결에서 빠집니다.

이렇게 하면 나중에 원본을 바꿀 일이 생겨도 바꾸는 자리가 한 곳입니다. 코드 여러 군데에 흩어진 선택을 찾아다니지 않아도 됩니다.

03

둘째, 같은 사건은 어느 경로로 들어와도 같은 함수를 지납니다

출하 하나가 화면에 따라 재고를 빼기도 하고 안 빼기도 했던 건은, 화면 두 개가 각자 저장 절차를 갖고 있었기 때문입니다. 한쪽이 재고 차감을 부르는 줄을 빠뜨렸습니다.

고침은 빠진 줄을 넣는 것이 아니라, **각자 갖고 있던 저장 절차를 없애는 것**이었습니다. 어느 화면으로

들어오든 출하는 하나의 같은 자리를 지나고, 그 자리가 재고 차감을 부릅니다. 화면은 값을 모아 넘기는 일만 합니다.

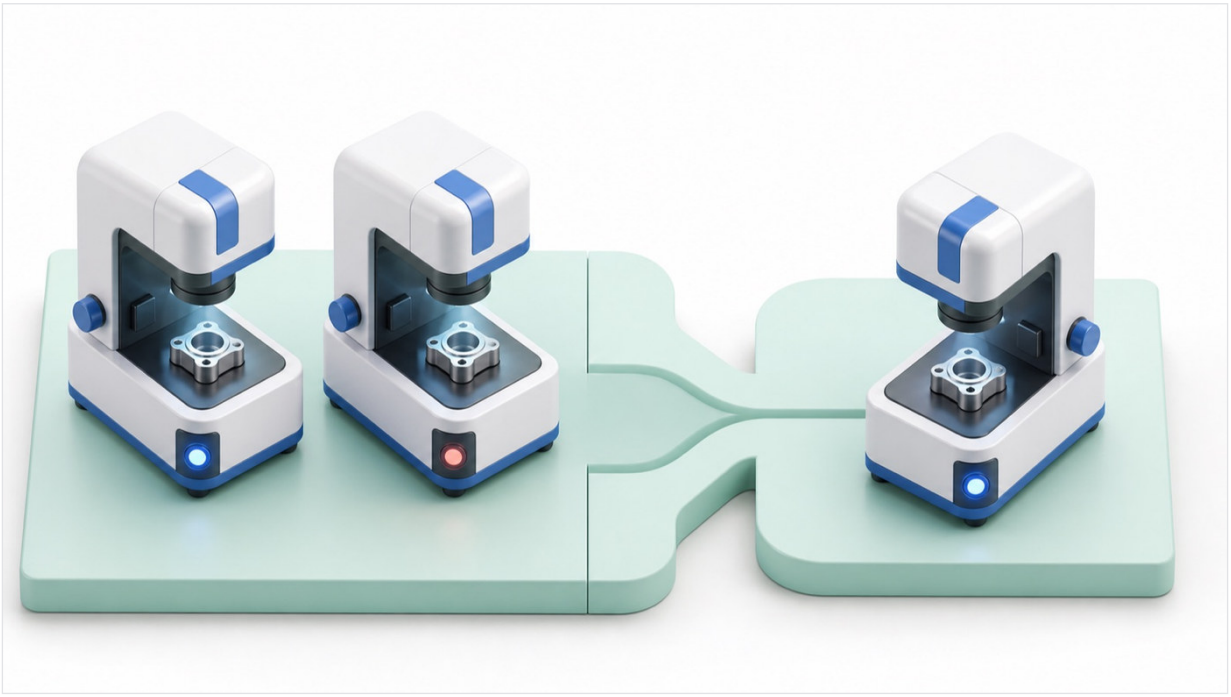
여기에 판정을 하나 붙였습니다. 같은 사건을 두 경로로 넣어 결과를 대조하는 시험입니다. 화면이 늘어나도 이 시험은 자동으로 같이 늘어납니다.

04

셋째, 판정하는 도구를 둘로 두지 않습니다

이번에 제일 뜨끔했던 것은 결함 자체가 아니라 검사 쪽이었습니다. 표 사이 연결 규칙을 보는 검사가 연결 대상을 엉뚱한 표에서 찾다가, 실제로는 양쪽에 다 있는 항목을 "없다"고 판정하고 그 규칙을 꺼 버렸습니다.

검사가 자기 판단으로 검사를 끈 것입니다. 그리고 그 검사는 그 사실을 통과로 보고했습니다.



같은 대상을 두 도구가 판정하면 서로 다른 답이 나옵니다. 하나로 합치면 답도 하나입니다.

원인은 같은 것을 판정하는 자리가 두 군데 있었던 것이었습니다. 하나로 합쳤습니다. 판정하는 도구가 둘이면 둘은 반드시 다른 말을 하는데, 다르다는 사실 자체를 알려 주는 자리가 없습니다.

같은 이유로 저희는 판정과 제안에 같은 계산기를 쓰게 합니다. "이건 문제입니다"라고 말하는 쪽과 "이렇게 고치면 됩니다"라고 말하는 쪽이 서로 다른 계산을 하면, 제안대로 고쳐도 판정이 안 풀리는 일이 생깁니다.

05

넷째, 값을 바꾸는 자리와 설명하는 자리에 같은 요구를 겁니다

선언 파일에 적어 넣은 설정 3건이 실제 등재는 0건이었던 일이 있었습니다. 값을 반영하는 쪽은 그 항목을 요구했는데, 승인하는 쪽은 요구하지 않았습니다.

그래서 **값은 조용히 바뀌고 설명은 조용히 사라졌습니다.** 나중에 그 값이 왜 그런지 물어보면 답할 사람이 없습니다.

지금은 두 자리가 같은 목록을 요구합니다. 설명이 없으면 값도 안 바뀝니다. 불편한 규칙인데, 반년 뒤에 "이건 왜 이렇게 돼 있죠"를 겪어 보면 이 불편이 훨씬 쌉니다.

06

다섯째, 못 건 제약은 못 걸었다고 적습니다

재고 전기 기록에 중복이 생기지 않게 제약을 걸려다가 못 걸었습니다. 이미 운영 중인 데이터에 중복이 있으면 그 제약을 거는 순간 **배포가 그 자리에서 실패**합니다. 그리고 정의 문서가 그 유일성을 선언한 적이 없어서, 저희 판단으로 강제할 근거도 없었습니다.

이럴 때 선택지는 셋입니다. 몰래 걸고 사고를 기다리거나, 안 걸고 잊거나, **못 걸었다고 적어 두는 것**입니다. 세 번째를 골랐습니다. 판정 목록에 "이 제약은 아직 걸려 있지 않다"가 항목으로 들어가 있고, 도입하는 회사마다 데이터 상태를 먼저 보게 되어 있습니다.

기능 목록에 없는 이야기라 팔 때는 손해입니다. 다만 도입 뒤에 배포가 멈추는 것보다는 낫습니다.

07

여섯째, 숫자에 이름을 정확히 붙입니다

값은 맞는데 이름이 틀려서 사람을 잘못 이끄는 경우가 있습니다. 설비 평균 고장 간격이 그랬습니다.

고장이 한 번도 없었던 설비는 계산할 고장 간격이 없습니다. 그래서 관측 기간을 고장 건수로 나누되 분모가 0이면 1로 두는 방식을 쓰고 있었는데, 이렇게 나온 값은 **평균이 아니라 하한선**입니다. "적어도 이만큼은 된다"는 뜻이지 "평균이 이만큼"이 아닙니다.



하한값 하나가 평균 자리에 앉으면 정렬 순서가 실제와 어긋납니다. 표시된 항목이 제자리를 벗어나 있습니다.

이름을 안 붙이고 평균 자리에 그대로 앉혀 두니 결과가 뒤집혔습니다. **고장이 한 번도 없던 설비가 고장이 한 번 난 설비보다 낮게 정렬**됐습니다. 이 순서를 그대로 보전 우선순위로 쓰면, 멀쩡한 설비를 먼저 손보고 실제로 위험한 설비를 뒤로 미룹니다.

값을 바꾸지는 않았습니다. 그 값이 무엇인지를 이름에 적었고, 정렬에서 하한값과 평균값을 섞지 않게 했습니다. 남은 질문도 값이 아니라 "고장이 한 번도 없던 달을 아예 빼고 셀 것인가" 하나로 좁아졌습니다. 그리고 순위 자체는 이 선택에 흔들리지 않아서, 보전 우선순위는 답을 기다리지 않고 지금 그대로 쓸 수 있습니다.

숫자가 틀리는 것보다 **맞는 숫자에 틀린 이름이 붙는 쪽**이 더 오래 안 잡힙니다. 값을 검산하는 사람은 있어도, 이 값이 무엇인지를 다시 묻는 사람은 드물기 때문입니다.

08

질문을 바꾸니 결정 4건 중 2건은 결정이 아니었습니다

이번 작업에서 방법 하나를 바꿨습니다. 결정이 필요한 항목을 쌓아 두고 답을 기다리는 대신, 목록을 열 때마다 질문을 이렇게 바꿔 물었습니다.

"이것이 정말 사람이 정할 일입니까." 그리고 **"답을 기다리는 동안, 어느 답이 나오든 지금 해도 되는 조치가 있습니까."**

결과가 뜻밖이었습니다. 사람이 정해야 한다고 적어 둔 4건 중 **2건은 결정이 아니었고, 그중 하나는 실물 손해**였습니다. 앞선 글의 발주 두 건이 그 하나입니다. 결정 대기 목록에 얌전히 앉아 있던 항목이, 사실은 답을 기다릴 필요가 없는 명백한 결함이었습니다.

같은 방식으로 다시 물었더니 저희가 적어 둔 전제도 무너졌습니다.

항목	적어 둔 전제	실제
공급사 규칙	두 규칙을 막고 있다	아무것도 안 막는다
시정조치	표준이 단일 등록부 요구	요구하지 않는다
검사 규격	사실상 같은 쌍둥이	겹치는 항목 0
입고 처리	둘 중 무엇을 고를까	질문이 성립 안 함

네 번째가 특히 그렇습니다. "입고를 창고 쪽에서 잡을까 회계 쪽에서 잡을까"를 오래 고민했는데, 전수로 열어 보니 한쪽에는 거래처 칸이 없고 다른 쪽에는 생산 입고 칸이 없었습니다. **고르는 질문 자체가 성립하지 않았습니다.** 답이 어려웠던 게 아니라 질문이 틀렸던 겁니다.

결정 대기 목록이 길어지면 그것을 처리 속도의 문제로 보기 쉽습니다. 저희 경험으로는 절반이 질문의 문제였습니다.

09

저희가 이번에 틀린 것도 적어 둡니다

규칙을 여섯 개 넣었다고 저희가 잘했다는 이야기로 읽히면 곤란해서, 같은 기간에 저희가 만든 오류를 함께 적습니다.

- **안전장치가 틀린 말을 화면에 띄웠습니다.** 사용자를 돕겠다고 넣은 안내 문구가 실제 상태와 다른 내용을 말하고 있었습니다. 없는 것보다 나쁩니다.
- **판정을 만들다가 시험 데이터 3행을 지웠습니다.** 검사를 만드는 작업이 검사 대상을 바꾼 겁니다.
- **저장이 안 되는 화면이 1,051개라고 잘못 셌습니다.** 실제로 실행해 보지 않고 코드만 읽어 짐작한 숫자였고, 돌려 보니 사실이 아니었습니다.

같은 형태의 오류를 바깥에서도 봤습니다. 검토를 받는 과정에서 "나눗셈에 0 방어가 없어 화면이 죽는다"는 지적을 받았는데, 실제로 8개 화면을 전부 돌려 보니 하나도 죽지 않았습니다. 값을 만드는 자리에서 이미 0을 걸러 내고 있었습니다. 지적한 쪽도 코드를 읽어 재구성해 짐작한 것이었고, 저희가 1,051개로 틀린 것과 정확히 같은 모양입니다. **짐작은 전문가가 해도 짐작입니다.**

세 번째는 특히 남겨 둘 만합니다. 코드를 읽어 짐작한 판단은 그럴듯한 만큼 위험합니다. 저희가 도입 검토하시는 분들께 "실제로 돌려 본 결과를 보여 달라"고 말씀드리는 근거가, 사실은 저희가 그렇게 틀려 봤기 때문입니다.

10

지금 상태

이 글을 쓰는 시점의 실측입니다.

- **자동 판정** — 47건
- **전체 검사** — 224건 전건 통과
- **시나리오** — 47건 중 47건

숫자가 좋아 보이지만, 앞의 글이 말한 결함 넷은 **이 검사가 전부 초록이던 상태에서 나왔습니다.** 그래서 통과 건수는 성적표가 아니라 지금까지 물어본 질문의 개수라고 보고 있습니다. 질문을 늘리는 일이 남은 일입니다.

아직 만드는 중인 소프트웨어이고, 완성됐다고 말할 단계가 아닙니다. 다만 이 여섯 규칙은 만들면서 배운 것이라, 어떤 시스템을 쓰시든 그대로 적용되는 이야기라고 생각해서 적었습니다.

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

원문 blog.vexplor.com/post/rules-so-numbers-dont-diverge
발행 2026년 8월 27일 · 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