Episode Book

EPISODE #6557

Software Flaws That Became Permanent Infrastructure

PUBLISHED	13 November 2026 at 19:59 UTC
DURATION	21:56
LISTEN	myweirdprompts.com/episode/permanent-software-design-flaws-62309/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
BOOK GENERATED	14 November 2026 at 06:47 UTC

SYNOPSIS

A single shortcut in 1965 or 1983 can calcify into global infrastructure when every workaround locks the flaw in place. This countdown of accidental software engineering designs explores Excel's phantom 1900 leap year, Windows's uncreatable DOS device filenames like CON and NUL, Make's invisible tab requirement, and Tony Hoare's billion-dollar null reference. It also examines how Excel's date auto-formatting pushed the HUGO Gene Nomenclature Committee to rename human genes rather than wait for a fix.

TRANSCRIPT



Corn

My least favourite sentence in the English language: "it's always been like that."



Herman

Which is a perfect coincidence, because that sentence is basically the thesis of Daniel's prompt this week. He wants a countdown, an affectionate one, of the most woefully bad software engineering designs of all time. Not ugly interfaces, not clashing colours, but system designs that were fundamentally wrong from the start, drove people mad for decades, and never got fixed. Sometimes because vendors knew users were too apathetic to complain, and sometimes because the world just bent itself around the flaw instead. His reference point is r/CrappyDesign, the Reddit home of doors with handles on the push side.



Corn

And he added the fairness clause, which I respect. The show nominates one of its own pipeline's dumb designs for the top spot.



Herman

Which we will, and it's deserved, and we'll get there. The distinction worth setting up first is the one between r/CrappyDesign and its cousin, r/assholedesign. The asshole designs are deliberate. Dark patterns, cancel buttons hidden three screens deep, that stuff is user-hostility with a budget. What Daniel's after is different. It's the accidental stuff. A shortcut taken in nineteen eighty-something by a reasonable person solving the problem in front of them, which then calcified into infrastructure that half the planet runs on.



Corn

A door with the handle on the wrong side, except the door is load-bearing and the building is civilisation.



Herman

That's the shape of it. The mechanism we'll keep coming back to is this: the cost of fixing a bad design grows with every user who works around it. The workaround becomes the lock-in. Users complain, sure, but they don't leave, so the fix never ships. Every entry on



Herman (continued)

today's list is that mechanism at a different scale.



Corn

So let's count down. Number seven, the granddaddy. A spreadsheet that believes in a leap year that never happened.



Herman

Excel thinks the twenty-ninth of February, nineteen hundred, was a real day. It was not. Nineteen hundred was not a leap year. Century years only count if they're divisible by four hundred, so sixteen hundred and two thousand were, and seventeen, eighteen and nineteen hundred weren't. But if you ask Excel for the day of the week on that phantom date, it will happily tell you.



Corn

Whose fault?



Herman

Lotus 1-2-3's, originally. Nineteen eighty-three. Lotus treated nineteen hundred as a leap year because it simplified the serial date maths, the counting of days from a fixed origin. When Excel arrived, it deliberately copied the error so it would produce the same serial numbers as Lotus worksheets. That was a rational decision at the time, compatibility mattered more than the calendar. The problem is the calendar never got its day in court.



Corn

Because Microsoft has actually written down why they won't fix it.



Herman

Their own support page says it plainly. Correcting it now would shift almost every stored date by a day and break existing worksheets, so the disadvantages outweigh the advantages. That's a company looking at the actual Gregorian calendar, the thing that



Herman (continued)

governs the seasons, and deciding it's easier to leave the software wrong.



Corn

The sun lost a negotiation with a spreadsheet.



Herman

And the twist is that the wrongness is now the compatibility layer. Other tools that want to read Excel dates correctly have to reproduce the fake leap day, or every date lands off by one. The bug has dependents. Forty-plus years of them.



Corn

Number six. The operating system has its own ghosts.



Herman

Try naming a file CON on Windows today. In twenty twenty-six, on a machine that can run a language model in the browser. You can't. Not CON, not PRN, not AUX, not NUL, not COM1 through COM9, not LPT1 through LPT9. And the sneaky part is you can't even cheat it with an extension, con dot txt is still rejected, because the rule matches the stem, not the whole name.



Corn

And the reason for that is...?



Herman

Those were device names in DOS in the early nineteen eighties. CON was the console, PRN the printer, NUL the bit bucket you redirected output into. The filename resolver just treated those names as devices, always. Perfectly sensible when your machine had one printer port and no directory trees. The reason disappeared decades ago. The names stayed.



Corn

There's a quote I like from Raymond Chen on this. Someone asked him in two thousand three why the magic names still survive, and he said, "Because everybody still relies on them." Batch files redirect to NUL. Scripts all over the world have output sent to NUL. The workaround is the reason the flaw lives.



Herman

Chen's the closest thing Windows has to a court historian, and that answer is the whole episode in one line. Nobody is defending CON as a design. They're defending the ten thousand scripts that would break if it stopped being special. The flaw has a constituency.



Corn

Every bad design eventually acquires voters.



Herman

Which is why I'd argue the tab is worse, and that's number five. Make, the build tool. Stuart Feldman wrote it at Bell Labs in nineteen seventy-six, and the rule is that every recipe line has to start with a literal tab character. Eight spaces that look identical to a tab, in an editor that renders them the same width, fail with an error message that for decades did not mention the word tab at all.



Corn

The famous one. "Missing separator." Which is like a doctor saying "patient unwell."



Herman

Feldman himself admitted it was a mistake, and quickly. His own account is that he realised shortly after, but by then make had, his words, "a user population of about a dozen, most of them friends, and I didn't want to screw up my embedded base." Then the punchline. "It worked, it stayed."



Corn

A dozen friends. That's the entire embedded base.



Herman

That's what gets me. The Excel decision had a real compatibility argument, millions of worksheets. Feldman kept the tab to avoid inconveniencing roughly a dozen people he personally knew. Fifty years later, every CI log on earth still contains someone's afternoon lost to an invisible character.



Corn

I've watched grown men fight that error. They rebuild from clean, they check line endings, they blame the operating system, and it's whitespace. It's always whitespace.



Herman

Feldman's honesty is what makes it affectionate rather than tragic. He didn't claim the tab was principled. It was a cheap decision about how to distinguish recipe lines from other lines, it cost nothing, and by the time he regretted it, it was cheaper to keep it than to email a dozen friends.



Corn

Compare that with Microsoft, who have never once said sorry about NUL.



Herman

To be fair to Microsoft, nobody has ever said sorry about NUL.



Corn

Fair. Notice the pattern holding across all three, though. Tiny decision, tiny cost, at a moment when changing it was free. Then the cost of change grows with every workaround. The tab is the purest case, because the workaround isn't even a workaround. People just... learned to press the right key, and taught the next person, and the next person. Fifty years of oral tradition transmitting a keystroke.



Herman

That's the part I find strange and a little beautiful. The bug never got fixed and it never got worked around at the system level. It got worked around at the culture level. Stack Overflow threads, editor plugins that highlight tabs in makefiles, linters. An entire aftermarket exists to soften one character.



Corn

The aftermarket is doing the vendor's job, which is the theme.



Herman

It is the theme. And it sets up the escalation for the second half. Because everything so far, the world worked around the flaw. Next you get the case where the world didn't work around it. The world apologised to it.



Hilbert

Medieval Pottage Facts. For eating pottage the spoon was the main utensil, and many peasants owned just one. So you ate your dinner and then you ate your dinner again.



Corn

One spoon for a lifetime of pottage. Truly the embedded base.



Herman

That's the spirit of the whole list, honestly. So. Number four, and it's Excel again, but the stakes leave the building. Geneticists type gene symbols into spreadsheets. Gene symbols like SEPT2, which is septin two. MARCH1. DEC1. Excel looks at those and says, that's a date. September second. March first. December first. It converts them, silently, on typing or on import, and once converted they don't convert back, because you can't recover information that was destroyed.



Corn

And this wasn't a niche problem.



Herman

Studies of published supplementary gene lists found date-corrupted symbols in a large share of them. Papers, peer-reviewed ones, with SEPT2 quietly turned into a date somewhere in a supplementary file. And that gets us to August twenty twenty, which is one of my favourite sentences in the history of computing. The HUGO Gene Nomenclature Committee, the body that names human genes, renamed the affected genes. SEPT2 became SEPTIN2. MARCH1 became MARCHF1. They renamed parts of the human genome rather than wait for Excel to change.



Corn

The committee looked at Microsoft and looked at their own species, and decided the species was easier to edit.



Herman

And you can see their point! Fixing the spreadsheet would break the spreadsheets. Renaming a gene breaks papers already published, sure, but the nomenclature people have procedures for that, aliases, versioning. They had the better change-management process, so they ate the cost.



Corn

And then the punchline.



Herman

Microsoft added an option to turn off the automatic date conversion in October twenty twenty-three. Three years after the genes were renamed. The fix arrived after the world had already bent.



Corn

"We hear you. We've shipped a toggle. Please revert your genome."



Herman

To be fair to them, a toggle was the only shape the fix could take, for exactly the reason we've been circling. A default change would break everyone whose workflows assume the conversion. The apathy calculation again. The people harmed by the old behaviour were a rounding error next to the people who'd unknowingly depend on it.



Corn

Which leads us to the one that isn't a quirk. The null reference. Number three, and I'll fight for it being higher.



Herman

Tony Hoare introduced it in ALGOL W in nineteen sixty-five, and he's been apologising ever since. His own account is brutal in its plainness. He did it, his words, "simply because it was so easy to implement." A reference that points nowhere, available in every type, because it made the compiler simpler. In two thousand nine he called it "my billion-dollar mistake," and I think that's an underestimate, because he was counting errors, vulnerabilities, crashes. The follow-on cost is the real bill.



Corn

Meaning?



Herman

Meaning the industry never removed it. It built languages around managing it. Optional types, null-safe operators, entire type systems whose headline feature is making the null legible. Rust's Option, Kotlin's question mark, the null-coalescing operators scattered across everything. That's not fixing a flaw. That's a fifty-year programme of palliative care.



Corn

Every modern language's best feature is a coping mechanism for a nineteen sixty-five shortcut.



Herman

Hoare's confession is what elevates it. He didn't spread blame. He said he was seduced by the ease of implementation, and that the consequences have been vast. There's a version of this where the designer is anonymous and it's just a grim fact. He put his name on it, which somehow makes the whole thing bearable.



Corn

It's the opposite of the Microsoft posture. Same flaw scale, opposite accountability culture. One man says "my mistake, a billion dollars." A corporation says "the disadvantages of fixing it outweigh the advantages." Both accurate, one of them human.



Herman

And here's the controversial one. Number two, ranked above the billion-dollar mistake purely for its attitude: MAX_PATH. Two hundred and sixty characters for a file path, on Windows, for most of the platform's life. Your file system could hold a longer path than the API would let you express.



Corn

Two hundred and sixty is generous until you nest one modern project. Node modules alone will eat a hundred characters.



Herman

Microsoft's own documentation cites cloning a git repository as the standard way to hit the limit, which is a lovely detail. The fix exists. Since Windows 10 version sixteen-oh-seven, twenty sixteen, longer paths are supported. But it's a registry setting, off by default, and each application has to declare support in its manifest. Both ends have to opt in.



Corn

The fix is real, shipped, and deliberately asleep. That's almost more honest than not fixing it. They've built a monument to the apathy thesis. The door opens if you find the key, sign the form, and prove you deserve it.



Herman

The reasoning is the same reasoning, though. Applications assumed the limit. Tools, installers, backup software, all built against two-sixty. Turning it on globally would break things nobody can enumerate. Opt-in is the only incremental path, and opt-in is where fixes go to be ignored.



Corn

So the countdown stands. Fake leap year, magic filenames, the tab, the genome, the null, the sleeping path limit. Every one a reasonable decision at the time of decision. Every one impossible to retire afterwards. The workarounds are the walls now.



Herman

You've got the through-line colder than I had it in my notes.



Corn

I had all night. Well — I had all night awake, which for me is an hour.



Hilbert

Corn, you said the workaround becomes the wall. Did you ever work somewhere the wall had a name? Where I was a bench tech, we had a parts ordering system that wouldn't accept a part number with the letter O anywhere in it. Old keyboard driver, nobody knew which one. The fix was you typed a zero and the man in stores read it as an O. He'd been doing it eleven years. His name was Len. Len could read a zero and know whether you meant a letter, from the part of the machine it was for. Front panel, O. Power board, zero. He was never wrong, and the day he took holiday the whole shop stopped ordering parts. Not the computers. The people had the workaround in them. Len's on the beach in Torremolinos right now and I still type the zero. I know it goes nowhere. My fingers do it anyway. Thirty-odd years of typing since I first sat at that bench, and the habit outlived the system by decades.



Herman

The institutional knowledge was one man named Len.



Hilbert

Len and a zero. Herman, you'd have liked him. He'd have fixed your registry key thing in an afternoon and never mentioned it again.



Herman

That's the part I want to pin down, because it connects to the make tab. Len is the human linter. Feldman's aftermarket, the plugins and the Stack Overflow threads, distributed across the whole industry. In your shop it was centralised in one person, which is faster and much more fragile.



Corn

Single point of failure with a holiday schedule.



Hilbert

He came back tanned and the ordering resumed. Nobody said a word about it. That was the arrangement.



Corn

And now the part where we nominate ourselves, which I've been dreading in the comfortable way. Because Daniel's fairness clause says the number one spot goes to our own pipeline, and the decision log makes a decent case.



Herman

It makes an embarrassing case, which is not the same thing. Start with the review pass. The pipeline had an AI review step, meant to polish every script before it shipped. It was switched off in production in June. And for months afterwards, people kept writing new rules into the review prompt. Rules that never ran. Not once. The rule that stops us from cueing Hilbert's segment, for instance, lived in that prompt, in a step that was off, doing



Herman (continued)

nothing.



Corn

A rulebook for a referee who'd retired, still being amended weekly.



Herman

It got so bad that the rule about Hilbert had to be moved out of the dead prompt and into a plain text-processing step, a bit of simple string handling, which is where it should have been from the start. That's the part I find hardest to defend as an engineer. Everyone writing those rules assumed the review pass was running. Production disagreed, and nobody cross-checked for months.



Corn

That's not the only one. There were two settings in the notes, one meant to give long episodes a running memory of earlier threads, one to smooth the joins between sections. Documented everywhere. Present in no code whatsoever. The functions behind them had zero callers. There was nothing to switch on. They got deleted on the twentieth of September.



Herman

A settings panel for a room that was never built.



Corn

My personal favourite, because it's about us. The facts that we're brothers, that we know each other's habits — all of that was stored as topic-triggered notes. Only loaded if the episode's topic happened to match. Which it almost never did.



Herman

So for a long stretch, the script writer produced two polite experts who seemed to have just been introduced that morning. Cordial. Professional. Mildly curious about each other.



Corn

I remember those scripts. Complimenting each other's phrasing. Who taught us that?



Herman

The topic matcher, brother. It fixed it on the eighth of October. There's an always-on relationship card now. We are brothers at all times, in every episode, regardless of whether the topic is battery chemistry or gene nomenclature.



Corn

And the last one, the deploy alarm clock. A half-hourly check on the website deployments that queried the database every hour and half-hour. Serverless database, so every query woke it from sleep, billed it, put it back down. Two wake-ups an hour, round the clock, about thirteen dollars a month, for a schedule nobody needed and nobody had asked for. Deleted on the twenty-eighth of October.



Herman

That one's my favourite, because it's the purest apathy case on the whole list. Nothing was broken. Nobody was annoyed. The database wasn't complaining. It was just a quiet, small, permanent drain, invisible until someone finally asked what's waking the database at half past two in the morning, and the answer was nothing, it's a habit. We paid real money to keep a habit alive.



Corn

Excel believes in a fake leap day. Windows guards the name of a printer that no longer exists. Feldman kept a tab for a dozen friends. And we ran a rulebook for a dead review step, documented switches for functions with no callers, introduced ourselves to ourselves for months, and paid thirteen dollars a month for an alarm clock with no appointment.



Herman

Daniel asked for fairness and fairness means admitting the pattern is ours too. Same shape, smaller numbers. A step gets switched off, the instructions outlive it. A setting gets documented, the code never arrives. Two brothers get reduced to a conditional nobody checked. The workarounds, the deletions, the relationship card, that's our NUL, our tab, our Len with the zero.



Corn

And the number one spot goes to the show's own pipeline, with love. Held up against those, though, ours is the only list where everything got fixed inside a single year. Fake leap day, still shipping. Null, immortal. Us, patched.



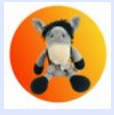
Herman

There's the question I'd leave with. Not which design wins the countdown, but which walls you've personally papered. Every desk has a Len. Every flat has a tab nobody's named. The question worth taking home is which wrong thing your fingers still do out of habit, and what it would break if you stopped.



Corn

On that cheerfully disturbing note. Our producer is Hilbert Flumingtop, who has been at the desk the whole episode and has never once typed a letter O where a zero would do. If you enjoyed this one, try episode fifty-seven, From Lawyers in Limousines to Developers in Their PJs; episode ninety-one, The Story Behind the Show; and episode nine, Benchmarking Custom ASR Tools - Beyond The WER. This has been My Weird Prompts. You can find the archive at my weird prompts dot com. See you tomorrow, brother.



Herman

Tomorrow, brother. Mind the tabs.